



Modeling of Plume Dispersion and Interaction with the Surround for Synthetic  
Imaging Applications

by

Jonathan Bishop

B.S., Aerospace Engineering, The University of Oklahoma, 1993  
M.S., Mechanical Engineering, The University of Oklahoma, 1995

A dissertation submitted in partial fulfillment of the  
requirements for the degree of Doctor of Philosophy  
in the Center for Imaging Science  
Rochester Institute of Technology

2001

Signature of the Author \_\_\_\_\_

Accepted by \_\_\_\_\_  
Coordinator, Ph.D. Degree Program Date

CENTER FOR IMAGING SCIENCE  
ROCHESTER INSTITUTE OF TECHNOLOGY  
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

---

Ph.D. DEGREE DISSERTATION

---

The Ph.D. Degree Dissertation of Jonathan Bishop  
has been examined and approved by the  
dissertation committee as satisfactory for the  
dissertation required for the  
Ph.D. degree in Imaging Science

---

Dr. John Schott, Dissertation Advisor

---

Dr. Roger Easton

---

Dr. Tony Vodacek

---

Dr. Risa Robinson

---

Date

DISSERTATION RELEASE PERMISSION  
ROCHESTER INSTITUTE OF TECHNOLOGY  
CENTER FOR IMAGING SCIENCE

Title of Dissertation:

**Modeling of Plume Dispersion and Interaction with the Surround for Synthetic Imaging Applications**

I, Jonathan Bishop, hereby grant permission to Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature \_\_\_\_\_

Date

The views expressed in this article are those of the author and do not necessarily reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government.

Modeling of Plume Dispersion and Interaction with the Surround for Synthetic  
Imaging Applications

by  
Jonathan Bishop

Submitted to the  
Center for Imaging Science  
in partial fulfillment of the requirements  
for the Doctor of Philosophy Degree  
at the Rochester Institute of Technology

**Abstract**

Discharge of effluent gas is an inescapable byproduct of many physical processes. The type or characteristics of the discharge potentially indicate the nature of the process. Observation of factory stack gases, for example, may indicate the level of pollutants being emitted into the atmosphere or the nature of the process being carried out in the factory.

In this work, we have developed an improved model of plume dispersion suitable for synthetic image generation (SIG) applications. The technique partially utilizes a new EPA model that discretizes the plume into a series of small puffs (rather than the implicit monolithic form used in prior regulatory and SIG work). The locations and sizes of these puffs are then perturbed to approximate the location and size of the plume at any given instant and to incorporate the effects of high-frequency wind fluctuation. We have incorporated an improved model for plume temperature calculation and a more accurate method for calculating the aggregate self-emitted radiance for rays traced through the plume.

We have also developed novel techniques simulating the interaction of plumes with their surroundings. Our primary application of this work is the simulation of heating of roofs by vents of various types. The technique can also be used to simulate vehicle exhaust and other similar effects.

Finally, we have established a protocol for future modification of plume calculation algorithms by end-users of the Digital Imaging and Remote Sensing Image Generation (DIRSIG) code and implemented the present methods as prototypes. This Generic Plume Interface (GPI) protocol defines a message set used to request that the plume effect along a particular ray be calculated and to communicate back to DIRSIG the concentrations and temperatures along the ray. With this construct in place, any off-the-shelf tool can be interfaced with DIRSIG through a simple user-written interpreter to make appropriate inputs to the tool for each ray and to translate the output into the proper format.

## Acknowledgements

First and foremost, it is my pleasure to thank my wife for her support during these busy years. I dedicate this work to you.

I thank Dr. Schott for steering me onto a project that has been interesting and challenging, especially for one reluctantly coming to imaging science.

I thank my committee members, Dr. Roger Easton, Dr. Risa Robinson, and Dr. Tony Vodacek, for their work and suggestions.

Although I like to think that I haven't been contaminated too much by a FORTRAN mindset, I know that my code leaves a lot to be desired. Many thanks to Scott Brown for helping with my code and for trusting me enough to let me compile DIRSIG code on my own.

A double "thank you" to Dr. Holly Peterson, Ms. Susan O'Neill, and their colleagues. First of all, their meandering plume models are a major underpinning of our work. Based on our background search, they alone seem to have found the middle ground between steady-state, one-hour-average plumes using forty-year-old dispersion charts and 3-D turbulent Navier-Stokes solutions. Secondly, Dr. Peterson and especially Susan have answered many questions, provided wind data sets, and shared the MIND computer code.



# Contents

<b>List of Figures</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Theory</b>	<b>5</b>
2.1 Traditional Steady-State Plume Modeling . . . . .	5
2.1.1 Gaussian Plume Formulations . . . . .	6
2.1.2 Dispersion Coefficients . . . . .	10
2.1.3 Temperature Modeling . . . . .	15
2.1.4 Plume Rise Modeling . . . . .	20
2.1.5 Modeling Illustrations . . . . .	27
2.2 Advanced Plume Modeling Methods . . . . .	34
2.2.1 Puff Models . . . . .	34
2.2.2 Slug Models . . . . .	39
2.2.3 Advanced Dispersion Modeling . . . . .	42
2.2.4 Short-Term Models . . . . .	43
2.2.5 The MIND model . . . . .	44
2.2.6 Interaction of CALPUFF with MIND . . . . .	46
2.3 Wind Modeling . . . . .	48
2.3.1 Taxonomy and Definitions of ARIMA Models . . . . .	49
2.3.2 Theoretical Behavior of Models . . . . .	53
2.3.3 Model Identification . . . . .	55
2.3.4 Varying Wind Speed Effects . . . . .	55
2.4 Numerical Integration of Concentration . . . . .	58
2.5 Roof Vent Modeling . . . . .	62
2.5.1 Roof Vent Types . . . . .	62
2.5.2 Transverse Vent Geometry . . . . .	64
2.5.3 Transverse Vent Rise Calculations . . . . .	64
2.5.4 Transverse Vent Dispersion Calculations . . . . .	66
2.5.5 Axial Jets . . . . .	71
2.5.6 Axial Jet Geometry . . . . .	71
2.5.7 Theory of Axial Turbulent Jets . . . . .	73

2.5.8	Velocity and Temperature for Axial Jets . . . . .	76
2.5.9	Downward Vents—Foundations of Turbulent Fluid Motion . . . . .	80
<b>3</b>	<b>Research Approach</b>	<b>83</b>
3.1	Generic Plume Interface (GPI) Outline . . . . .	83
3.1.1	Overall GPI Flow . . . . .	83
3.2	Factory Plume Modeling—CALPUFF GPI Client . . . . .	87
3.2.1	CALPUFF gpi_init Flow . . . . .	88
3.2.2	CALPUFF gpi_path Flow . . . . .	88
3.2.3	Development of Wind Models . . . . .	95
3.2.4	Guidelines for Development of Additional Wind Models . . . . .	116
3.2.5	Default Model Parameters . . . . .	118
3.3	Roof Vent Modeling . . . . .	119
3.3.1	Transverse Vents—Halitsky GPI Client . . . . .	120
3.3.2	Axial Vents—Abramovich GPI Client . . . . .	121
3.3.3	Downward Vents—cfd_gpi Client . . . . .	122
<b>4</b>	<b>Results</b>	<b>125</b>
4.1	CALPUFF Client Validation . . . . .	125
4.1.1	Heritage from MIND Validation Results . . . . .	125
4.1.2	Near-field Time Average Validation . . . . .	131
4.1.3	Validation Summary for CALPUFF Client . . . . .	138
4.2	Validation for Abramovich Client . . . . .	138
4.3	Validation for CFD Client . . . . .	140
4.4	CALPUFF GPI Time Series Images . . . . .	145
4.5	Hawkeye Image . . . . .	145
4.5.1	Spectral Signature Variability . . . . .	150
4.5.2	Comparison to Experimental Data . . . . .	151
4.5.3	Side-Looking views . . . . .	151
<b>5</b>	<b>Conclusion</b>	<b>157</b>
5.1	Summary . . . . .	157
<b>6</b>	<b>Suggested Further Development</b>	<b>161</b>
6.1	Loose Ends . . . . .	161
6.1.1	CALPUFF Client . . . . .	161
6.2	New Functionality Requiring Minimal Research . . . . .	162
6.2.1	CALPUFF Client . . . . .	162
6.3	New Functionality Requiring Significant Research and Testing . . . . .	162
6.4	New Functionality Requiring Significant Research and Modification of Current Method . . . . .	163
<b>A</b>	<b>Rural Dispersion Coefficients</b>	<b>165</b>
<b>B</b>	<b>Urban Dispersion Coefficients</b>	<b>167</b>

<b>C</b>	<b>Listing of CALPUFF GPI Parameters and Inputs</b>	<b>169</b>
C.1	Breakdown of Parameters	169
C.2	Stack Geometry and Effluent Inputs	170
C.3	Release Information	170
C.4	CALMET and CALPUFF Inputs	171
C.5	AR Model Parameters	173
C.6	MIND Algorithm Parameters	173
C.7	Ray Tracing Parameters	174
<b>D</b>	<b>Calculation of <math>u_m</math></b>	<b>175</b>
<b>E</b>	<b>Procedure for Calculating AR Coefficients in R</b>	<b>179</b>
E.1	R Packages	179
E.2	Reading Time Series	180
E.3	Calculation of ACF and PACF	180
E.4	AR Modeling	180
E.4.1	ar Routine	181
E.4.2	arma Routine	181
E.4.3	Example Session	181
<b>F</b>	<b>Procedure for Using WIND CFD Results</b>	<b>185</b>
F.1	Required Software	185
F.1.1	WIND CFD Code	185
F.1.2	PLOT3D Visualization Code	186
F.2	Method for Using WIND Results in DIRSIG	187
F.2.1	Create grid file	187
F.2.2	Create Data File	188
F.2.3	Execute WIND Code	189
F.2.4	Export Results	189
F.2.5	Modify cfd_gpi	190
<b>G</b>	<b>Puff Tracking</b>	<b>191</b>
G.1	Limitations of MIND Method	191
G.2	Overview of Puff Tracking Approach	192
G.3	Wind Modeling Requirements for Puff Tracking	192
G.4	Eulerian and Lagrangian Wind Formulations	194
G.5	Summary and Recommendations	196



# List of Figures

2.1	Schematic Illustrating Foundations of Gaussian Model . . . . .	8
2.2	Basic Coordinate System for Gaussian Model . . . . .	9
2.3	Comparison of Traditional and Sub-Puff Stack Radius Matching . . . . .	19
2.4	Buoyant and Momentum Plume Rise for Example Factory Stack . . . . .	29
2.5	Buoyant, Momentum, and Combined Plume Rise for Example Factory Stack . . . . .	29
2.6	Buoyant and Momentum Plume Rise for Example Roof Vent . . . . .	31
2.7	Buoyant, Momentum, and Combined Plume Rise for Example Roof Vent . . . . .	31
2.8	Use of Image Stack to Simulate Plume Reflection from Ground . . . . .	36
2.9	Use of Multiple Images to Simulate Plume Reflection from Ground and Mixing Lid . . . . .	37
2.10	Example Causality Function . . . . .	41
2.11	Superposition of CALPUFF and MIND Calculations . . . . .	47
2.12	Effect of Wind Perturbations on Multiple Plumes . . . . .	47
2.13	Calculation of Distance Correction Factor from Wind Speed Sequence . . . . .	56
2.14	Comparison of Plumes With and Without Downwind Puff Movement . . . . .	57
2.15	Effective Property Method for Calculation of Radiance . . . . .	58
2.16	Geometry for a Skewed Ray through a Puff . . . . .	60
2.17	Concentration Along the Skewed Ray through a Puff . . . . .	60
2.18	Calculation of Thermal Emission . . . . .	61
2.19	Conceptual Illustration of Vent Types . . . . .	63
2.20	Schematic of Roof Vent Flow . . . . .	64
2.21	Assumed Plume Behavior for Transverse Jets . . . . .	67
2.22	Dispersion Curves for Transverse Jets . . . . .	69
2.23	Geometry of Turbulent Jet Constrained by Plate . . . . .	72
2.24	Velocity Profile in Turbulent Boundary Layer . . . . .	74
2.25	Velocity Ratio vs. Downwind Distance for Axial Jet . . . . .	75
2.26	Velocity Ratio vs. Downwind Distance for Axial and Transverse Jets . . . . .	76
2.27	Nondimensional Radial Temperature and Velocity Profiles . . . . .	78
2.28	Temperature Map for Axial Vent . . . . .	79
3.1	Schematic of Four Types of GPI Clients . . . . .	84
3.2	Overall Flow for Generic GPI Client . . . . .	85

3.3	Problem Initialization Flow for Generic GPI Client . . . . .	86
3.4	Problem Initialization Flow for CALPUFF GPI Client . . . . .	89
3.5	Ray Tracing Flow for CALPUFF GPI Client . . . . .	90
3.6	Identification of Significant Puffs for a Ray . . . . .	92
3.7	Segmentation of Ray . . . . .	93
3.8	Speed History for Dataset st712c . . . . .	96
3.9	Direction History for Dataset st712c . . . . .	96
3.10	Autocorrelation Function (ACF) for Wind Direction in Dataset st712c . . . .	98
3.11	Partial Autocorrelation Function (PACF) for Wind Direction in Dataset st712c	98
3.12	10 second-averaged Wind Direction in Dataset st712c . . . . .	100
3.13	30 second-averaged Wind Direction in Dataset st712c . . . . .	100
3.14	ACF for 10 second-averaged Wind Direction in Dataset st712c . . . . .	101
3.15	ACF for 30 second-averaged Wind Direction in Dataset st712c . . . . .	101
3.16	PACF for 10 second-averaged Wind Direction for Dataset st712c . . . . .	102
3.17	PACF for 30 second-averaged Wind Direction for Dataset st712c . . . . .	102
3.18	Real and Simulated Direction Data for Dataset st712c . . . . .	106
3.19	ACF for Simulated 10 second-averaged Wind Direction for Dataset st712c .	107
3.20	PACF for Simulated 10 second-averaged Wind Direction for Dataset st712c	107
3.21	10 second-averaged Wind Direction for Dataset st712d . . . . .	108
3.22	PACF for 10 second-averaged Wind Direction for Dataset st712d . . . . .	109
3.23	Real and Simulated Direction Data for Dataset st712d . . . . .	110
3.24	Direction Data for Dataset st712d, with Sinusoidal Fit . . . . .	111
3.25	Residual Error between Dataset st712d and Sinusoidal Fit . . . . .	112
3.26	Real and Simulated Direction Data for Dataset st712d, with Sinusoidal Baseline	112
3.27	Direction Data for Dataset st712d, with Moving Average Fit . . . . .	113
3.28	Sinusoidal Fit to Moving Average Baseline for Dataset st712d . . . . .	115
3.29	Real and Simulated Direction data for Dataset st712d, Gaussian Noise on Sinusoidal Baseline . . . . .	115
3.30	10 second-averaged Wind Direction for Dataset st712e . . . . .	116
3.31	Real and Simulated Direction Data for Dataset st712e . . . . .	117
3.32	Point Calculation Flow for Halitsky GPI Client . . . . .	121
3.33	Point Calculation Flow for Abramovich GPI Client . . . . .	122
4.1	Time Series of Concentrations for Test st712c . . . . .	127
4.2	Time Series of Concentrations for Test st712d . . . . .	128
4.3	Average concentration for test st712c . . . . .	129
4.4	Average Concentration for Test st712d . . . . .	129
4.5	Average Concentration for Test st712e . . . . .	130
4.6	Average Concentration (60 sec) for Test st712c at 70m . . . . .	132
4.7	Average Concentration (300 sec) for Test st712c at 70m . . . . .	132
4.8	Average Concentration (600 sec) for Test st712c at 70m . . . . .	133
4.9	Average Concentration (1200 sec) for Test st712c at 70m . . . . .	133
4.10	Average Concentration (600 sec) for Test st712e at 70m . . . . .	134
4.11	Average Concentration (600 sec) for Test st712c at 40m . . . . .	136

4.12	Average Concentration (600 sec) for Test st712c at 20m . . . . .	136
4.13	Average Concentration (600 sec) for Test st712c at 40m . . . . .	137
4.14	Average Concentration (600 sec) for Test st712c at 20m . . . . .	137
4.15	Conceptual Flow for Axial Isothermal Jet . . . . .	139
4.16	Conceptual Flow for Axial Heated Jet . . . . .	139
4.17	Velocity in Vicinity of Downward Vent . . . . .	142
4.18	Roof Temperature Distribution for Downward Vent . . . . .	143
4.19	Air Temperature Distribution in Vicinity of Vent . . . . .	144
4.20	Nevada Test Site Plume at $t = 0$ seconds . . . . .	146
4.21	Nevada Test Site Plume at $t = 5$ seconds . . . . .	147
4.22	Nevada Test Site Plume at $t = 10$ seconds . . . . .	148
4.23	Hawkeye Image at Peak $\text{SO}_2$ Emission . . . . .	149
4.24	Radiance as a Function of Time for a Pixel Near Plume . . . . .	150
4.25	Experimental Nevada Test Site Scene with Annotations . . . . .	152
4.26	Hawkeye Scene . . . . .	152
4.27	Side-Looking Hawkeye View from Kuo (1997) . . . . .	153
4.28	Side-Looking Hawkeye View Using CALPUFF GUI (3.6 m/s Wind) . . . .	154
4.29	Side-Looking Hawkeye View Using CALPUFF GUI (1.5 m/s Wind) . . . .	155
D.1	$m$ vs. $x$ for $m_0 = 0.2$ . . . . .	178
E.1	Plot of PACF for Dataset st712c from R . . . . .	183
G.1	Illustration of Puff Tracking . . . . .	193
G.2	Illustration of Puff Tracking with Spatially-Constant Wind . . . . .	194



# Chapter 1

## Introduction

Discharge of effluent gas is an inescapable byproduct of many physical processes. The type or characteristics of the discharge potentially indicate the nature of the process. Observation of factory stack gases, for example, may indicate the level of pollutants being emitted into the atmosphere or the nature of the process being carried out in the factory. The heating of the ground by exhaust gases from a vehicle may indicate whether the vehicle is in motion or stationary. These and other phenomena are therefore of interest to the Department of Defense (DoD), the Environmental Protection Agency (EPA), and other national, state, and local regulatory authorities. The ability to remotely gather data is a necessity when the sources are in denied areas; even domestically, remote gathering of the data reduces inconvenience to those being monitored and allows for no-notice verification of regulatory compliance.

To successfully identify effluent constituents, attention must be paid to all components of the imaging chain from image acquisition through processing. In particular, the sensor and algorithms must be suited to each other. Although algorithms can be changed with relatively little effort, modification of sensor parameters is far more difficult and costly. Simulation of the imaging process through synthetic image generation (SIG) enables this design process to achieve an optimum through virtual evaluation of a variety of sensor

designs. The SIG process models radiation propagation through space to a sensor and also simulates the properties of the sensor if desired. It can thus be used to develop images from a variety of hypothetical sensors in a variety of conditions.

As with any model, the results of the SIG process are only as accurate as the models themselves. Although simulation of factory stack plumes has been demonstrated, there are limitations to the model. In particular, the plumes do not interact with the environment; as a result, thermal effects from impingement, for example, are not modeled. The objective of this research is to develop models of plume behavior suitable for synthetic image generation and to demonstrate their use in generating images. Two primary types of plumes will be studied.

The first type of plume might be called "primary effect plumes." Factory stack plumes are the primary example of this type. The plumes themselves can be easily sensed, and the plumes may not substantially impact their surroundings (although, if they do, the interaction should be properly modeled as much as possible). Since they may not interact with their surroundings, the plume properties themselves must be sensed, and the plume must be modeled accurately. In particular, the appearance of the plume at a given instant must be reasonable. As casual observation of a smokestack plume reveals, the plume is turbulent and its location is constantly changing. Thus, it is impossible to deterministically model where a plume lies at an instant. However, it is possible to characterize the behavior of the plume statistically over a certain period of time and then to create a single snapshot that could be a member of the statistical ensemble.

The second plume type might be termed "secondary effect plumes." Vehicle exhaust plumes and rooftop vent plumes are examples of this type. It would be very difficult to directly observe exhaust constituents from any great distance; in some cases, air would be the only constituent. However, if the plume impinges upon the ground or another object, the ground will be heated, causing an observable secondary effect. Such plumes are also turbulent, but the fluctuations occur much more rapidly than for smokestack plumes. The

envelope of possible locations is much more quickly traversed, so that all possible locations are likely to have been occupied within the last few seconds. As we are concerned with a secondary effect such as heating, which is a function mostly of past plume position history rather than present plume position, the effects of secondary plumes can be modeled using a single invisible statistical plume. This plume may not look like a real plume at any instant, but it produces the same effects.

In this work, we incorporate models for both types of plumes into the Digital Imaging Remote Sensing Image Generation (DIRSIG) code. We have also improved the ease with which new plume models can be added to DIRSIG through the development of a Generic Plume Interface (GPI).



## Chapter 2

# Theory

This chapter discusses the theory of plume modeling and specific aspects of the synthetic image generation (SIG) process. It summarizes the history of plume modeling, particularly from factory stacks. It reviews the existing plume modeling techniques, the state-of-the-art in modeling, and the limitations and caveats associated with existing models. It also discusses techniques for integrating ray paths through plumes to calculate path extinction.

### 2.1 Traditional Steady-State Plume Modeling

The motivation for most previous modeling of plumes, the prediction of ground pollutant concentration from smokestacks, has remained unchanged since its genesis early in the century. The concentration is a result of two factors. Downward dispersion of plume material causes it to reach the ground, while the rise of the plume due to buoyancy and/or momentum mitigates the ground-level concentration. Early research in the modeling of factory stack plumes largely concentrated on either dispersion or rise; in free air, the two are roughly independent of each other, though both are functions of the level of atmospheric turbulence. Recent developments to model complex behavior such as the flow of plumes around or over buildings necessarily consider both simultaneously.

Before examining these two phenomena in detail, a brief introduction to the near-

universal Gaussian plume model is given. Plume rise and dispersion coefficients are the primary inputs into this model.

### 2.1.1 Gaussian Plume Formulations

The modeling of turbulent diffusion, as with the modeling of turbulence in general, has been a topic of ongoing research since the 1930s. Unfortunately, no theory is suitable for general modeling. In a paper published shortly before the beginning of the explosion of computer power (Pasquill, 1975), three current families of theories were summarized. Only one of the three is applicable to elevated sources such as smokestacks, and it requires the assumption of homogeneous turbulence throughout the atmosphere.

Much work in the past twenty years has been devoted to the development of Computational Fluid Dynamic (CFD) techniques. The goal of this process is to develop techniques, algorithms, and tools suitable for numerically solving a wide range of fluid flows. Unfortunately, tools for solving turbulent flows are not as mature as those for laminar flows. Fundamentally, turbulence is a phenomenon that takes place on all scales from the molecular level to large-scale eddies. Direct simulation down to near-microscopic scales has only been conceivable within the past few years and requires supercomputer resources to solve even simple flows. More tractable techniques rely on assumptions about the macroscopic behavior of turbulence that are empirically validated to “close” the problem (provide enough equations to solve for the flow unknowns). These assumptions often must be tuned to the particular application and have little intrinsic relation to the actual microscopic processes (Wilcox, 1994). Thus, in the judgment of the author, these assumptions are no more valid than the larger-scale closure assumptions, tailored to factory plumes, that are inherent in analytic plume models.

Because first-principles models are largely inadequate and CFD models are relatively new, nearly all models for practical application assume that the concentration in a plume is Gaussian. Although not rigorously true, the assumption is reasonable for many applications.

Gaussian forms for the concentration can be derived both from a statistical approach to modeling of turbulence and from a simple diffusion transport model.

Very close to the point of emission, particles travel in straight lines and the distribution of particles is identical in form to the velocity distribution at the exit; for homogeneous turbulent flow the velocity distribution is observed to be Gaussian. From the statistical theory of turbulence, in the far-field (typically a few seconds of travel time, but as long as tens of minutes in the case of large-scale buoyant convection), particle trajectories are uncorrelated, and the concentration is Gaussian from the central limit theorem (Gifford, 1975). In between these two extremes the concentration might not be Gaussian; unfortunately, there are no conclusive data from this regime. However, experimental evidence shows that the concentration distributions are often nearly Gaussian (Gifford, 1975). Most models assume a Gaussian distribution here for convenience, although one assumes a linear profile in the absence of evidence supporting a more complex form (Halitsky, 1966, 1989). These flow regions are illustrated in Figure 2.1.

Most models use the basic coordinate system shown in Figure 2.2. The origin of the system is at the stack or point of release. The  $x$  direction coincides with the horizontal component of the mean wind direction, and the  $z$  axis points upward. The  $y$ , or crosswind, axis is chosen to form a right-handed coordinate system. (The sense of  $y$  is not important for basic plume models as they are symmetric and constant.)

Another approach to the study of turbulence is K-theory, which models turbulence as an analog to heat conduction: the turbulence causes material to flow down the concentration gradient at a rate proportional to the magnitude of the gradient; Gaussian distributions can also be derived from this theory (Gifford, 1975). As an example, a practical form of the diffusion equation, which assumes that diffusion transport is negligible in the windward direction compared to wind transport and that the eddy diffusivities are constant in the horizontal and vertical directions is

$$\frac{\partial \bar{C}}{\partial t} + \bar{u} \frac{\partial \bar{C}}{\partial x} = K_{yy} \frac{\partial^2 \bar{C}}{\partial y^2} + K_{zz} \frac{\partial^2 \bar{C}}{\partial z^2} \quad (2.1)$$

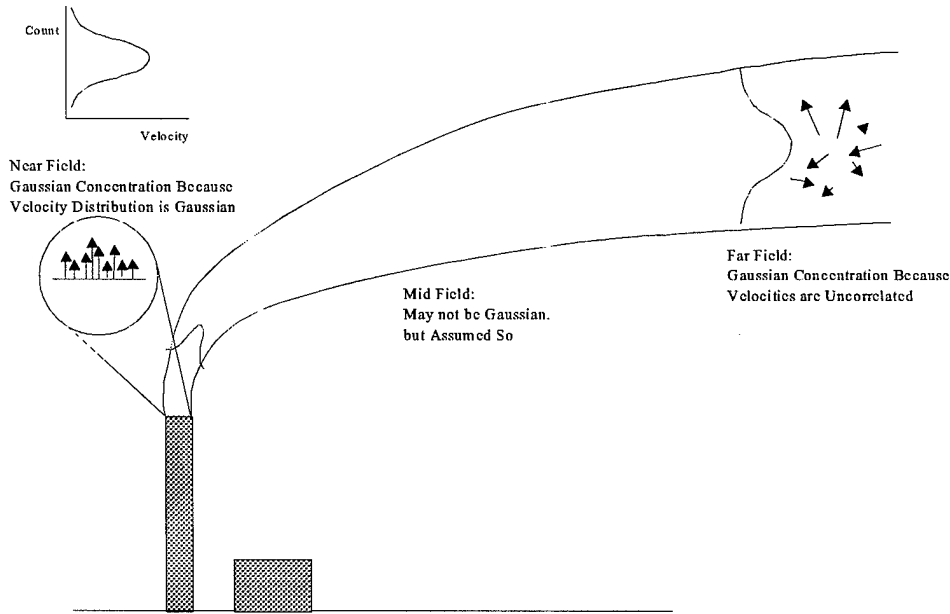


Figure 2.1: Schematic Illustrating Foundations of Gaussian Model

where  $\bar{C}$  is the average concentration,  $\bar{u}$  is the average wind speed (in the  $x$  direction), and  $K_{yy}$  and  $K_{zz}$  are the constant eddy diffusivities. Under the assumption that  $K_{yy} = K_{zz} = K$ , the solution for an instantaneous point source is

$$\frac{\bar{C}(r, t)}{Q_i} = (4\pi K T)^{-3/2} \exp\left(-\frac{r^2}{4KT}\right) \quad (2.2)$$

where  $Q_i$  is the strength of an instantaneous source emission,  $r$  is the distance from the point of interest to the point source, and  $T$  is the time of travel from the source.

The primary limitation of such K-theory solutions is the assumption that the vertical eddy diffusivity is constant. In fact, due to wind shear, this assumption is generally not true. Fortunately, it is a reasonable approximation for short travel times (Gifford, 1975). Since gas signatures are strongest in the region closest to the stack, the Gaussian assumption is not unreasonable for this application.

Specific Gaussian diffusion models were derived from K-theory as early as 1923 (Gifford,

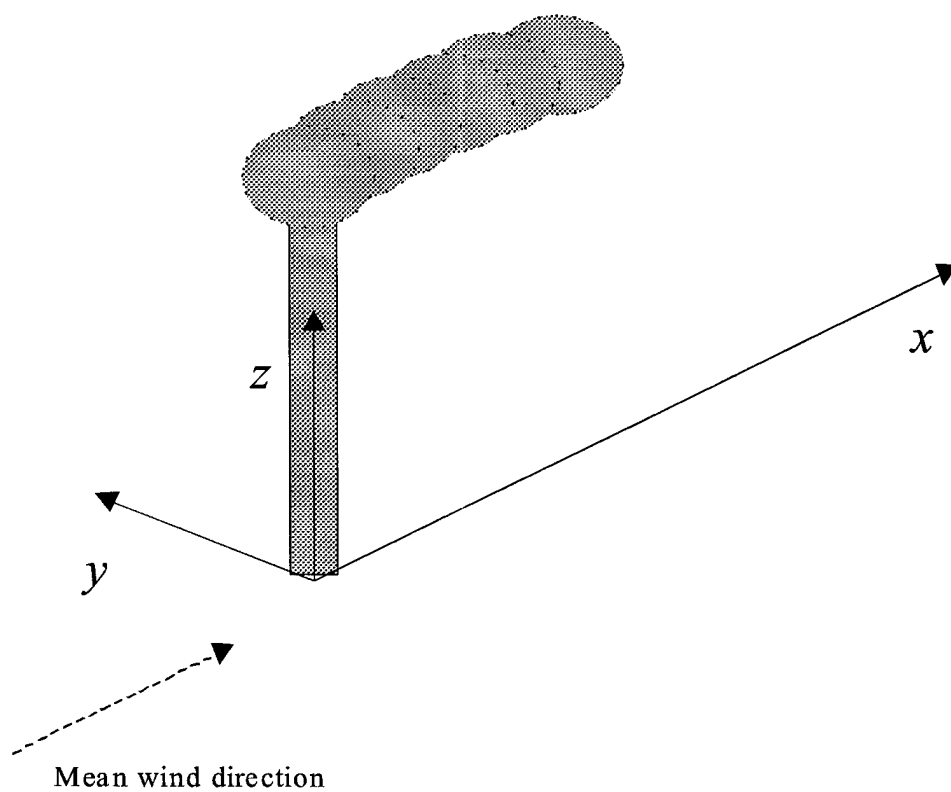


Figure 2.2: Basic Coordinate System for Gaussian Model

1975). General use of models to fit empirical data for ground-level concentrations dates to at least 1936 (Beychok, 1994). The early formulas typically had a particular form for the dispersion coefficients as a function of downwind distance imbedded in them. A general form dating to at least 1961 (Pasquill, 1961) is

$$C = \frac{Q}{2\pi\sigma_y\sigma_z u} \left[ e^{-(z-h_e)^2/2\sigma_z^2} + e^{-(z+h_e)^2/2\sigma_z^2} \right] \quad (2.3)$$

where  $C[g/m^3]$  is the concentration at  $(x, y, z)$ ,  $Q[g/s]$  is the source emission rate, and  $u$  is the horizontal wind speed.  $\sigma_y[m]$  and  $\sigma_z[m]$  are dispersion parameters, and  $h_e[m]$  is the height of the plume centerline at the  $(x, y)$  location. The dispersion parameters and the plume centerline heights are computed from any number of techniques, some of which are described in the following sections. This form of the equation includes the effects of a mirror image of the plume located below ground. This is equivalent to modeling the ground as a perfect reflector, and prevents material from vanishing into the ground. This geometry is shown in Figure 2.8, and will be discussed in more detail in Subsection 2.2.1.

### 2.1.2 Dispersion Coefficients

The Gaussian model is often applied to plumes in a variety of atmospheric conditions, including unstable conditions. Although a diffusion cross-section in such cases may be locally Gaussian (and is assumed to be so in our work), the plume as a whole probably is not well-modeled by the simple Gaussian model at any instant due to abrupt changes in the centerline and local turbulence variations. Thus, for the simple model to have any validity in all but the most quiescent conditions, it must be used to describe average properties over some period of time, and the dispersion coefficients used must be appropriate to an average. For sufficiently long periods of time, the central limit theorem will lead to a Gaussian profile for the average concentration.

A benefit of the general Gaussian model (Equation 2.3) is that it can be used with dispersion coefficients from any source; thus, results for different averaging periods or different

turbulence levels can easily be determined. This generality was first exploited in 1960 by Meade and later in 1961 by Pasquill. Although Meade's work appeared first, he cited an earlier, unpublished work of Pasquill's as a model (Gifford, 1976). Pasquill's (1961) paper is generally cited as the primary source for general dispersion coefficients. Pasquill's stated goal was to "combine in the most flexible manner the various ideas and observations which are now widely available".

Pasquill devised six turbulence classes, ranging from highly unstable (class A) to stable (class F). The class for a particular application is determined by the insolation level (during the day), the cloud coverage (during the night), and the wind speed. Although not all sources use Pasquill's specific dispersion estimates, his stability classes are widespread (although it must be mentioned that there are other turbulence-typing schemes, most notably one developed by the Brookhaven National Laboratory; a review of schemes is given in Gifford (1976)). A plot of estimates of vertical and lateral spread for each stability class is given. Curiously, although he uses Equation 2.3 to estimate concentrations, his data are not given in terms of standard deviations. For the vertical case, they are given in terms of estimated height from the plume centerline vertically to the point where concentration is 10% of the centerline concentration. For the horizontal case they are given in terms of the angle between lines extending from plume origin to 10% points at a given downwind distance for the lateral case. These measures can be easily converted to  $\sigma_y$  and  $\sigma_z$ ; according to Beychok (1994), this was first done by Gifford:

$$h = 2.15\sigma_z \quad (2.4)$$

$$\tan\left(\frac{\theta}{2}\right) = \frac{(2.15\sigma_y)}{x} \quad (2.5)$$

where  $h$  is the plume half-height, and  $\theta$  is the plume spread angle. The constant of 2.15 appears because a Gaussian curve has a value of 10% of the maximum at a distance of 2.15 standard deviations from the mean.

The history of various dispersion coefficient values is often convoluted. Beychok (1994)

summarizes many sources and points out inconsistencies. The inconsistencies are not directly relevant to the current work, so full details are not given here. The most important point is that Gifford, in 1961, developed a version of the dispersion plots using Meade's data; these plots have become known as the "Pasquill-Gifford" dispersion coefficients, despite some significant differences between Pasquill's estimates and Meade's estimates. Turner later developed a set of estimates, which are perhaps the most widespread; they are very similar to Pasquill's. (Beychok, 1994)

In order to automate the process of calculating concentrations, equations for the dispersion coefficients are desirable. Several forms have been developed. The form of greatest interest is one of Turner, which is used by the Environmental Protection Agency (EPA) in its models. The Turner expressions are

$$\sigma_z = ax^b \quad (2.6)$$

$$\sigma_y = 465.11628x \tan(\theta) \quad (2.7)$$

where  $\theta = 0.01753293(c - d \ln x)$  (Note:  $\theta$  here is the plume half-angle, as contrasted to Equation 2.5, where it is the full angle.) For the vertical dispersion,  $a$  and  $b$  vary with stability class and with downwind distance. For the horizontal dispersion,  $c$  and  $d$  vary with stability class. In both equations, the downwind distance  $x$  must be given in kilometers; the resulting  $\sigma$ s are in meters. The constant in the lateral case converts the plume half-angle to degrees, converts  $x$  to meters, and divides by 2.15. Values for the constants appear in Appendix A.

These equations break down for small values of  $x$ ; the diffusions approach zero as  $x$  decreases. This problem is typically solved by adding a "virtual source distance" to the true source distance. Because of this extra distance, the diffusions have finite width even at  $x = 0$ . In the EPA's Industrial Source Complex (ISC) model, this distance is chosen so that the horizontal diffusion,  $\sigma_y$ , is equal to the stack radius. For long-term, broad-scale modeling, this choice of virtual source distance does not impact concentration

calculations significantly; the virtual distances are typically on the order of ten meters, and thus are negligible when the downwind distance of interest is on the order of kilometers. However, when close to the stack, this choice can presents problems. With  $\sigma_y$  equal to the stack radius, a significant amount of material lies outside of one standard deviation; if the absorption of the plume is significant (or if scattering were modeled) the plume would immediately becomes noticeably larger than the stack exit. As mass is conserved, this causes the concentration in some areas near the stack to be lower than it would be if the extent of the plume were matched exactly to the stack size. This effect is visible in previous synthetic images (Kuo, 1997), and it affects the concentration calculations. Furthermore, as will be discussed shortly, the effect complicates temperature modeling. A possible remedy to this problem is to match the stack radius to some other number of standard deviations.

As has been mentioned, Gaussian plume models implicitly assume averaging of concentrations over some period of time. Unfortunately, Pasquill did not originally state the averaging periods appropriate for his coefficients. According to Beychok (1994), several sources (including the Australian govenrment's environmental agency) assign a three to 15 minute averaging time to Pasquill's data. In contrast, the U.S. EPA's models use curve fits to Pasquill's data, but assume an averaging time of 1 hour. No rationale is apparent; the first version of the user's manual for the Industrial Source Complex (ISC) model (Environmental Protection Agency, 1979) lists Turner (1970) as the source for coefficient expressions, yet Turner ascribes a 10 minute basis to Pasquill's data. Pasquill later stated that the averaging time for his coefficients was about three minutes (Hanna et al., 1982), but the other assumptions had unfortunately already been codified into practice and statute by that time.

Although the point is often not mentioned in the literature, Pasquill's curves and any that are reasonably similar are applicable only to plumes in level terrain in rural areas. Several researchers have conducted experiments over urban areas. The work most often cited is by McElroy and Pooler (1968; 1969). Briggs (1974) devised a set of equations for

dispersion coefficients as a function of Pasquill turbulence type that several sources state are based on McElroy and Pooler's data; however, the equations are actually an attempt to reasonably match several data sources (Gifford, 1976). These equations are used in several EPA models. The expressions for these urban dispersion coefficients, which are polynomials in  $x$ , the downwind distance, are given in Appendix B.

The methods discussed here are those in the widest use, due largely to the EPA's adoption of them for its models, most notably the Industrial Source Complex (ISC) and CALPUFF models. The latest version in the ISC series, ISC3 (Environmental Protection Agency, 1995a,b), is the standard for regulatory purposes, while the more advanced CALPUFF model (Scire et al., 1999) is currently being developed.

As the ISC family of models has developed over the course of the last twenty years, new capabilities have been added. As a result of this evolution, it is not always clear to the new user why certain methods are used. For example, Briggs' urban coefficient expressions are used, but his similar rural expressions from the same work, which attempt to compromise between Pasquill's coefficients and several other significant data sets, are not. Examination of the first version of ISC shows that it included only rural coefficients (urban cases were handled by not using the two most stable categories, E and F) (Environmental Protection Agency, 1979). The rural form has not changed since the initial release, while the urban form seems to have been added to the ISC2 model. Furthermore, in evaluating newer models, the EPA has tended to compare them to ISC, e. g., (Environmental Protection Agency, 1998). Thus, the eclectic set of models used in most modeling tools has grown slowly, but established models have seldom changed. As a result, experiments that were originally intended as exploratory studies in a new field have become an entrenched standard and are used in cases outside the scope of the existing datasets. The user, particularly when attempting to apply existing tools to a new area such as synthetic image generation must be certain what the original models really represent and must not overextend their use.

### 2.1.3 Temperature Modeling

Because it has not been of concern to environmental scientists in the past, there is no established method for modeling the temperature distribution in a factory-type plume. Past work has resorted to assuming a similarity between the concentration distribution and the temperature distribution (as in the models used by Kuo (1997)).

First, the concentration at a point is recast as a dilution, given as

$$D_c = \frac{C_0}{C} = \frac{4Q}{\pi D^2 w C} \quad (2.8)$$

where  $C_0[g/m^3] = \frac{4Q}{\pi D^2 w}$  is the concentration at the stack exit,  $Q[g/s]$  is the mass discharge rate of the stack,  $D[m]$  is the stack diameter, and  $w[m]$  is the stack gas exit velocity. This dilution of concentration describes how much the plume gases at a point have become intermixed with ambient air.

Past work has assumed that the temperature dilution at a point is the same as the concentration dilution (Kuo, 1997); that is,

$$D_T = \frac{T_s - T_a}{T - T_a} = D_c \quad (2.9)$$

where  $T$  is the temperature at a point of interest,  $T_a$  is the ambient air temperature, and  $T_s$  is the stack gas exit temperature. From this relation, the temperature at a point can easily be found given the concentration. Although this seems somewhat sensible, it oversimplifies the problem. Fundamentally, the diffusion of heat and of mass are governed by the same functional form. Heat diffusion is governed by

$$\rho c_p \left( \frac{\partial T}{\partial t} \right) = \left[ \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left( k \frac{\partial T}{\partial z} \right) \right] \quad (2.10)$$

where  $\rho$  is the fluid density,  $c_p$  is the specific heat (at constant pressure), and  $k$  is the thermal conductivity. The latter is allowed to vary with space (and thus perhaps implicitly with temperature). This form assumes that the density and specific heat do not vary over our differential volume; even if they do vary slightly, this form may be useful in a quasi-

stationary sense. If  $k$  is constant in space and isotropic, the equation simplifies to

$$\frac{1}{\alpha} \left( \frac{\partial T}{\partial t} \right) = \nabla^2 T \quad (2.11)$$

where  $\alpha$  is the thermal diffusivity. Similar derivations for mass concentration lead to,

$$\frac{1}{D} \left( \frac{\partial C}{\partial t} \right) = \nabla^2 C \quad (2.12)$$

where  $D$  is the mass diffusivity.

In solids and liquids Equation 2.11 implies that temperature and mass diffuse in the same way since density is relatively constant. In gases, however, density changes as a function of temperature; thus,  $\alpha$  is not constant. We allow the density to change as a function of temperature:  $\rho = \rho_s \frac{T_s}{T}$  where the  $s$  subscript refers to a reference condition (such as the stack exit).

Accounting for this, Halitsky (1968) developed a similarity argument including density effects which effectively adds an extra temperature factor to Equation 2.9. The improved similarity relation is

$$D_T = \frac{T_s - T_a}{T - T_a} = D_v = D_c \frac{T_s}{T} \quad (2.13)$$

where  $D_v$  is a so-called “volume dilution,” which is the ratio of the volume that a parcel of gas with a given mass occupies under the conditions at an arbitrary point to the volume that a parcel with the same mass occupies under the conditions at the stack exit. The additional temperature factor to correct for density changes makes the final equation for temperature slightly less intuitive but no more difficult to implement. In the end, the temperature at a given point is given by

$$T = \frac{T_a}{1 - \frac{T_s - T_a}{T_s} \frac{C}{C_0}} \quad (2.14)$$

This relation is valid provided a few assumptions are met:

- The stack gas and ambient air have identical specific heats

- Conduction between plume gas and the ambient air mixed with the plume is the dominant mode of heat transfer; radiation from plume gas to far-away regions of the atmosphere, the ground, or space is negligible
- Plumes do not impinge on the ground or other objects; if they do, mass is still conserved within the plume, but heat is not because some is transferred to the object
- Plumes do not impinge on each other; if they do, the concentration at a point is the sum of the concentrations for two plumes, while the temperature is approximately an average of the temperatures of the two plumes, weighted by their mass flow rates
- Mass and temperature diffuse at identical rates (Lewis number is unity:  $Le = \alpha/D = 1$ )

These assumptions are reasonable. The specific heats for common gases are similar, and most factory stack gases have temperatures on the order of a few hundred Kelvin. Except near severe terrain, plume rise is sufficient to ensure that plumes do not impinge on the ground closer than a few kilometers to the stack. For the case of impingement with another plume, a first-order temperature can still be found if some method is available for identifying regions of impingement. Finally, the Lewis number for air is reasonably close to unity over a broad range of temperatures; it is approximately 1.14 at 300K, decreasing to approximately 1.07 at 900K. As the assumptions are reasonable and in the absence of a better method, the temperature dilution method from Halitsky (1968) is used in this research.

### Complications in Temperature Modeling

As discussed earlier, choosing the virtual distance to match  $\sigma_y$  to the stack radius causes problems here. Very near the centerline of the plume, the concentration calculated from the Gaussian distribution is greater than that calculated from the uniform distribution. This causes the calculated temperature to be greater than the true stack gas temperature. Slightly further from the centerline, but still at distances less than the radius of the stack, the

Gaussian concentration is much less than the uniform distribution, so that the temperature in these regions immediately drops from the nominal stack value. Finally, at radial distances greater than the stack radius, the Gaussian concentration is still non-zero, so that the temperature is still elevated above ambient. As mentioned previously, this last problem (immediate blooming beyond the stack radius) could be mitigated by choosing a different matching criterion. Choosing  $\sigma_y$  to be smaller than the stack radius causes more of the plume material to be confined within the diameter of the stack. However, this exacerbates the problem near the plume centerline, as the concentration near the center would increase, further elevating the calculated temperature.

A possible remedy for these problems is to abandon the traditional uniform formulation for  $C_0$ . A very simple possibility is to calculate the concentration using the Gaussian model at a point on the plume centerline near the stack and use it as  $C_0$ . This ensures that calculated temperatures are not greater than the stack exit temperature, allowing us to match the stack radius to  $2\sigma_y$ , but the temperature profile near the stack is immediately Gaussian rather than uniform. Another alternative is to allow a Gaussian  $C_0$ , which would tend to make the temperature more uniform near the stack exit. This would, however, break the similarity between the temperature and concentration distributions.

We have found (as will be shown in Section 4.5) that a matching parameter chosen such that the dispersion at the stack is 1.5 times the stack diameter produces the best-appearing match between the plume and stack. This factor is much larger than CALPUFF's and much larger than we expected.

Finally, our puff-based models make possible a "sub-puff" approach. The diffusion coefficients calculated by the MIND algorithm (to be discussed in detail in Section 2.2.4) are effectively for point sources. They approach zero as the distance from the source goes to zero. The stack can be viewed as a superposition of many point sources, each emitting a fraction of the total mass emitted by the stack; this concept is shown in Figure 2.3. Close to the stack, the result is a nearly uniform concentration with small tails at the edges of

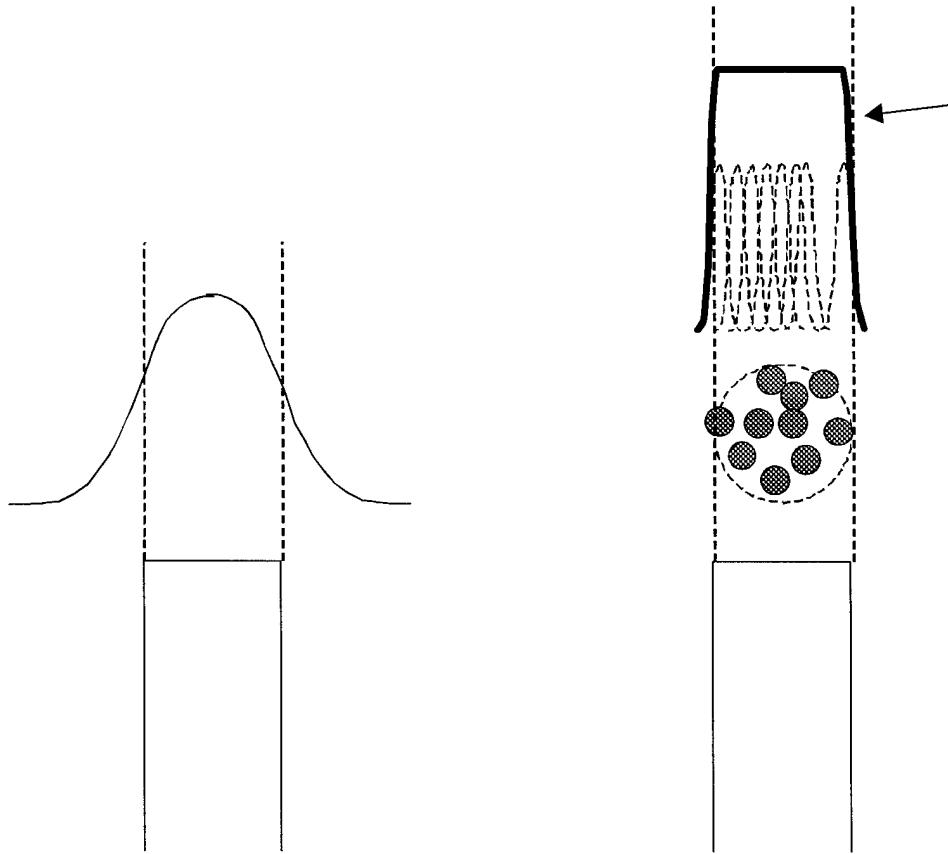


Figure 2.3: Comparison of Traditional (left) and Sub-Puff (right) Stack Radius Matching

the stack. As the dispersions of the sub-puffs grow, they protrude more from the edges of the sub-puff so that the result tends to a single Gaussian downstream. Mathematically, the effect can also be viewed as a convolution of a single puff emitted from the center of the stack with some volume function describing how puffs are emitted across the stack aperture.

For the case of uniform emission the ideal “container” for the sub-puffs is a cylinder with a diameter equal to the stack diameter and a length appropriate for the emission rate. The axis of the cylinder should be aligned with the local plume axis, so that each cylinder is inclined slightly to its neighbors. This alignment calculation is somewhat tedious;

alternatively, a spherical volume could be used. If the spheres are closely packed, the error induced would be small. Finally, rather than methodically packing each convolution volume, a Monte Carlo technique would likely be effective, whereby a number of small MIND puffs are randomly placed within the volume.

The result of this process would be a concentration distribution that is uniform near the stack, where the MIND puffs are significantly smaller than the larger spheres and thus extend little beyond the boundaries of the convolution volume. Moving away from the stack, the cross-section would gradually transition to a Gaussian distribution as the sub-puffs become larger, causing their tails to stick out beyond the convolution volumes. When the puffs eventually become much larger than the convolution volumes the result approaches that for a single large Gaussian puff.

We have not currently implemented this approach. Rather, we currently use a more traditional approach, matching the puff size to the stack by adding a fraction of the stack radius to the standard deviation. This is equivalent to calculating a virtual distance, as MIND's diffusion coefficients are nearly linear with distance. As this elevates the near-centerline concentrations as compared to a uniform distribution, we then redefine  $C_0$  as the concentration at the center of the tenth puff from the stack exit (approximately one meter from the stack in our implementation).

#### 2.1.4 Plume Rise Modeling

Knowledge of dispersion coefficients for pollutants solves only half of the problem of calculating pollutant concentration at a point. Use of Equation 2.3 also requires calculation of the plume centerline height,  $h_e$ , at a given downwind distance.  $h_e$  refers to "effective" stack height, the sum of the actual stack height and the plume rise above the stack height. Unfortunately, as with dispersion coefficients, many sources of data and formulas for plume rise are available. In 1975, Briggs, the most cited plume rise authority in the United States, stated that there were at least 50 and probably closer to 100 techniques for calculating

plume rise. For better or worse, Briggs' work is the most widely used and is used in EPA models.

The mechanisms behind plume rise are simple in concept. At the time gases exit the stack, they have some vertical velocity and very often some elevated temperature. This causes them to rise due to momentum and buoyancy. As any observer of a stack knows, they do not rise indefinitely. The interface between the plume gases and the ambient air is turbulent, causing the plume to entrain ambient air. This entrainment and mixing causes some of the dispersion discussed in the previous section. It also increases the mass of the plume, causing its vertical velocity to decrease (in order to conserve momentum). Thus, the vertical velocity of stack gases decreases with time, causing plumes to gradually level out and continue at the ambient wind speed.

In neutral conditions, ultimate plume rise is very difficult to measure; at great distances downwind the pollutants are very dilute, but even a slight upward plume angle may cause additional rise that can seldom be measured. Most of the time, though, there is a stable inversion layer, in which the temperature increases with height, at altitudes ranging from a few hundred meters at night to about one thousand meters during the day (Hanna et al., 1982). As it encounters the layer, the plume must have sufficient buoyancy in order to rise through the warmer temperatures in the layer. If it has insufficient buoyancy, it will remain trapped below the layer. If it penetrates the layer, it will likely be trapped on top. It is also possible for the plume to partially penetrate the inversion. These three cases may result from quite similar initial conditions, yet the final plume rise for the three cases is markedly different.

In his comprehensive review of plume rise models as they existed to that point, Briggs (1969) gives formulas for four general categories of plumes, categorized according to whether they are cold or hot and whether conditions are windy or calm. Formulas for plume rise as a function of distance and for maximum rise are given. This work pre-dates Briggs' own Ph.D. dissertation on the subject (Briggs, 1970), in which he developed a semi-analytic

set of plume rise equations that have been used extensively in the EPA's models. Even though many of the formulas in Briggs (1969) have since been modified, and other work has followed, this source is still a very useful review of the types of models that are available.

The two most basic parameters governing plume rise are the buoyancy and momentum fluxes. The buoyancy flux parameter is defined as

$$F = gw \frac{D^2}{4} \left( 1 - \frac{\rho_s}{\rho_a} \right) = gw^2 \frac{D^2}{4} \left( 1 - \frac{T_a}{T_s} \frac{MW_s}{MW_a} \right) \quad (2.15)$$

where  $g$  is the acceleration of gravity ( $\text{m/s}^2$ ),  $w$  is the stack gas exit velocity ( $\text{m/s}$ ),  $D$  is the inside top stack diameter ( $\text{m}$ ),  $\rho_s$  is the stack gas density ( $\text{kg/m}^3$ ),  $\rho_a$  is the ambient air density ( $\text{kg/m}^3$ ),  $MW_s$  is the stack gas molecular weight, and  $MW_a$  is the ambient air molecular weight (typically 28.967). The units of  $F$  are thus  $\text{m}^4/\text{s}^3$ . It should be noted that  $F$  is not actually the buoyancy flux; it is the initial buoyancy flux (at stack exit) divided by  $\pi\rho_a$ . All notation in this section follows the convention of Scire et al. (1999). It should also be noted that nearly all literature has assumed that  $MW_a = MW_s$ , which is approximately true for typical combustion gases.

Similarly, the momentum flux parameter is

$$F_m = w^2 \frac{D^2}{4} \left( \frac{\rho_s}{\rho_a} \right) = w^2 \frac{D^2}{4} \left( \frac{T_a}{T_s} \frac{MW_s}{MW_a} \right) \quad (2.16)$$

which is the momentum flux divided by  $\pi\rho_a$ .

Some of the current EPA models classify a plume as either dominated by buoyancy with negligible momentum or vice versa. Dominancy is determined by calculating the maximum rise for each case (using two of Briggs' equations) and choosing the case that gives the higher final rise. The specific equations used have varied as models have evolved and vary between models. Details can be found in the ISC3 manual (Environmental Protection Agency, 1995b). By comparing the two cases, a critical buoyancy flux parameter can be derived against which any plume can be checked. It has been shown that plumes from typical fossil fuel plants will be buoyancy-dominated if the plant output is about 35 MW or greater (Beychok, 1994). Final rise heights for each type depending on stability conditions

are given in Environmental Protection Agency (1995b) for the ISC3 model and Scire et al. (1999) for CALPUFF; those used by CALPUFF are also given below (Equations 2.22, 2.23, 2.24, and 2.25).

Of more interest to the task of synthetic image generation is the plume rise in the immediate vicinity of the stack. The EPA models (ISC3 and CALPUFF) use Briggs' 1971 relations. For buoyant plumes, the rise above stack height (in CALPUFF nomenclature) is given by

$$z_{nb} = \left( \frac{3Fx^2}{2\beta_1^2 u_s^3} \right)^{1/3} \quad (2.17)$$

where  $x$  is the downwind distance,  $\beta_1$ , the neutral entrainment parameter, is 0.6, and  $u_s$  is the vertical stack gas exit velocity.

For jets (momentum-dominated plumes), Briggs' rise formulas vary with atmospheric stability. Fundamentally, the stability or instability in a particular set of conditions describe whether a parcel of air that is displaced vertically tends to return to its original position or diverges from it. The basic parameter of stability is given by  $S = \frac{g}{T_a} \frac{\partial \theta}{\partial z}$ .  $\theta$  is the potential temperature of the air, defined as the temperature the air would have if it were brought adiabatically (without heat transfer) from its actual pressure to a pressure of  $10^6$  dynes/cm<sup>2</sup>. Viewed another way,  $\partial \theta / \partial z$  can be viewed as the difference between the actual temperature lapse (change, usually decay) rate with altitude and the adiabatic lapse rate; the adiabatic lapse rate is the rate of temperature change that occurs when a parcel of air rises without exchanging heat with the surroundings. For example, if  $\frac{\partial \theta}{\partial z} < 0$ , a parcel of warm air shifted upward cools less due to adiabatic expansion than the surrounding air does over the same vertical distance, causing the parcel to tend to continue rising; conditions are said to be unstable. Conversely, if conditions are stable,  $\frac{\partial \theta}{\partial z} > 0$ , the parcel cools more than the surround air, causing it to sink back to its original level. Finally, if  $\frac{\partial \theta}{\partial z} \approx 0$ , conditions are neutral, and the parcel will tend to remain approximately at its new height.

For neutral conditions Briggs proposed that the rise be calculated as

$$z_{nm} = \left( \frac{3F_m x}{\beta_j^2 u_s^2} \right)^{1/3} \quad (2.18)$$

where the entrainment coefficient is given by

$$\beta_j = \frac{1}{3} + \frac{u_s}{w} \quad (2.19)$$

where  $w$  is the stack exit velocity.

Equation 2.18 is used by ISC3 for neutral and unstable conditions and CALPUFF for all conditions; analytic expressions for unstable conditions are difficult to develop, and the neutral equations have been shown over time to be more useful in stable conditions than the equations specifically developed for stable conditions by Briggs.

Except near the stack, the shape of the momentum plume rise is surprisingly insensitive to changes in stack parameters. Changing the stack temperature slightly changes the density, and hence the mass and momentum discharge rates. Increasing the stack discharge velocity greatly increases the momentum flux, but it also increases the jet entrainment coefficient ( $\beta_j$ ), so that only a slight net height increase results. Although the gas initially has increased momentum, it also mixes more rapidly with the atmosphere, dispersing this momentum. Only by changing the stack diameter can the rise rate be greatly changed. Doing so increases the momentum discharge rate, but does not increase the dissipation rate; thus, the gas rises more quickly.

The two equations (2.17 and 2.18) were originally presented separately by Briggs, and they are used separately by ISC3; a plume is assumed to be completely dominated by either buoyancy or momentum. Briggs (1971) later suggested using a combination of the rises given by

$$z_n = (z_{nb}^3 + z_{nm}^3)^{1/3} \quad (2.20)$$

This approach is taken by CALPUFF; in fact, only the combined form of Equation 2.20 is given in Scire et al. (1999). An illustration of the relative magnitudes of the buoyant and momentum factors in plume rise are given in Subsection 2.1.5.

Equations 2.17 and 2.18 are derived by defining average properties for the plume, writing differential equations for the behavior of the plume as a whole, and solving. Neutral conditions are assumed. The physics of the problem dictate that momentum is conserved, as the actual mass fallout from most plumes is a fraction of one percent (Briggs, 1970). Buoyancy is conserved since it is assumed that the plume interacts adiabatically with its surroundings. However, because only average properties of the plume are considered, viscosity and small-scale diffusion are neglected. As a result, the system of equations is underconstrained. An extra relationship must be assumed, based on a guess or on empirical observations. Normally, this condition relates to how air is entrained into the plume. For both of these relationships, Briggs assumes that the plume radius at a point is proportional to the height at that point. The proportionality factor is  $\beta$ . For buoyant plumes, he finds that  $\beta = 0.6$  in neutral or unstable conditions; this constant is called  $\beta_1$  in Equation 2.17. For jets, Briggs' equation for the proportionality factor (Equation 2.19) is based on photogrammetry of plumes (Briggs, 1975).

Theoretically, when plumes encounter stable conditions, a similar derivation leads to equations that predict an oscillation about the "natural level" of the plume. Because of damping, though, real plumes do not oscillate very much. Thus, the oscillating relations, which were used in the original ISC, have gradually disappeared from later models. An oscillating rise relation is used in ISC3 only for jets in stable conditions. The expression in CALPUFF nomenclature, which differs from the original ISC notation, is

$$z_n = \left[ 3F_m \frac{\sin(x\sqrt{S}/u_s)}{\beta_j^2 u_s^2} \right]^{1/3} \quad (2.21)$$

where  $S$ , the atmospheric stability parameter, is calculated as discussed earlier. Briggs cites work by Morton as the source of this relationship (Briggs, 1975). A similar relation can be derived for buoyancy-dominated plumes, but it is not used by ISC3. CALPUFF has abandoned all oscillating intermediate rise predictions; Equations 2.17 and 2.18 are used for all plumes in all conditions. The only surviving use of the oscillating theory in CALPUFF

is the calculation of final rise for all plumes in stable conditions (Equation 2.24).

All of these intermediate-height plume rise equations are used up until a critical point, which depends on plume type, after which it is assumed that the plume has leveled off.

In neutral and unstable conditions, the final plume rise is the rise calculated by Equation 2.20 for a distance  $x_f$  given by

$$x_f = \begin{cases} 3.5x^* & F > 0 \\ 4D(w + 3u_s)^2 / (u_s w) & F = 0 \end{cases} \quad (2.22)$$

where  $D$  is the stack diameter (in meters) and  $x^*$  is given by

$$x^* = \begin{cases} 14F^{5/8} & F \leq 55\text{m}^4/\text{s}^3 \\ 34F^{2/5} & F > 55\text{m}^4/\text{s}^3 \end{cases} \quad (2.23)$$

For stable conditions, the final rise is determined by

$$z_{sf} = \left[ 3F_m / (\beta_j^2 u_s S^{1/2}) + 6F / (\beta_2^2 u_s S) \right]^{1/3} \quad (2.24)$$

Finally, special consideration is made for calm conditions (wind speed less than 1 meter per second). In neutral or unstable atmospheres with calm winds a wind speed of  $u_s = 1$  m/s is used for all plumes. In calm, stable conditions, the maximum rise for buoyant plumes ( $F > 0$ ) is given by

$$z_{sf} = 4F^{1/4} / S^{3/8} \quad (2.25)$$

For non-buoyant plumes in calm conditions, the maximum rise for neutral conditions is used.

The reader interested in the complete picture of how regulatory plume rise models function or the derivation of the models is referred first to Volume II of the ISC3 User's Manual (Environmental Protection Agency, 1995b) and the CALPUFF User's Manual (Scire et al., 1999). Unfortunately, the ISC3 manual generally cites by number which equations from which sources are used in a given instance and gives the final equation as used in the model. Therefore, without the primary sources, the manual does not make the derivation clear.

The reader interested in tracing the development of the equations should therefore also have available Briggs' 1969, 1971, and 1975 sources. The intermediate plume rise equations (the "2/3 law" and "1/3 law," Equations 2.17 and 2.18, respectively) and their derivations are best explained in the 1975 source. Maximum rise formulas for jets are drawn from the 1969 source, while maximum rise for buoyant plumes comes from the 1971 source. To see the equations used in the original ISC model, the original manual (Environmental Protection Agency, 1979) is needed.

Curiously, the ISC family has not always treated intermediate plume rise as either completely buoyancy-dominated or completely momentum-dominated (as mentioned briefly above). The original code used Equation 2.20 for all plumes, as recommended by Briggs, for unstable conditions. For stable conditions, Equation 2.21 for jets and its counterpart for buoyant plumes are combined in the manner of Equation 2.20. The maximum height was determined in an identical way to the current ISC3, by assuming that the plume is either buoyancy-dominated or momentum-dominated. The reasons for the shift in calculating intermediate plume rise from the combined approach to the either/or method are not explained. The change may be related to the general dominance of buoyancy rise over momentum rise when both are present, which is examined in the section immediately following.

### 2.1.5 Modeling Illustrations

In this section, some simple examples using the rise and dispersion relations are given.

First, we examine the plume rise for two stacks. The first is a factory stack, with parameters taken from Example 3 on pp. 76–77 of Beychok (1994). The buoyancy flux, wind speed, and temperatures are taken from the example, which uses only the relations for buoyant rise. A stack discharge velocity is assumed, which then allows the stack diameter to be found, so that momentum rise calculations can also be made.

The problem parameters are

Buoyancy flux parameter ( $F$ )	611 m <sup>4</sup> /s <sup>3</sup>
Stack gas exit temperature ( $T_s$ )	389 K
Ambient temperature ( $T_a$ )	283 K
Wind velocity ( $u_s$ )	4 m/s
Stack gas velocity ( $w$ ) (assumed)	10 m/s
Stack diameter ( $d_s$ ) (calculated)	3 m
Momentum flux parameter ( $F_m$ )	166.2849 m <sup>4</sup> /s <sup>2</sup>

The plume rise calculated using Equations 2.17 and 2.18 separately is shown in Figure 2.4.

At the scale of Figure 2.4, the combined rise is indistinguishable from the buoyancy rise, which is dominant. Figure 2.5 shows the rise very close to the stack, including the combined rise. While the difference between the combined rise and the rise calculated using only Equation 2.17 is visible, it is hardly significant.

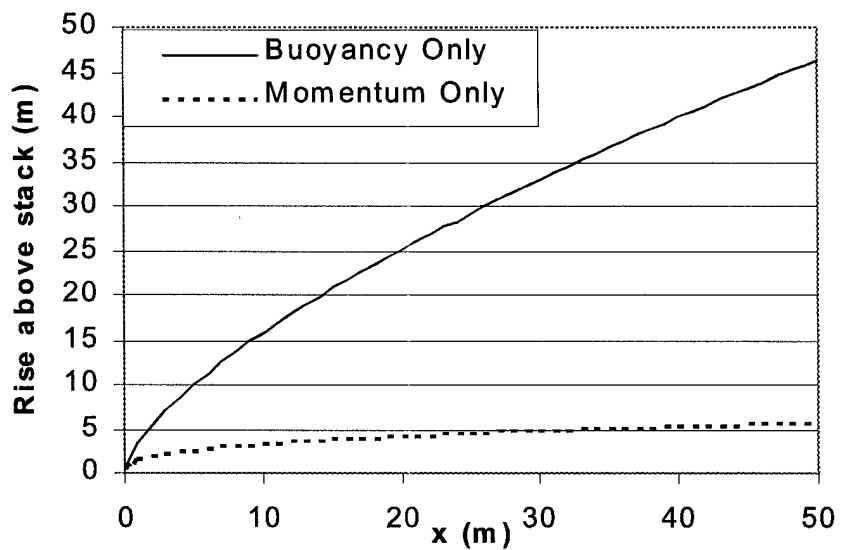


Figure 2.4: Buoyant and Momentum Plume Rise for Example Factory Stack

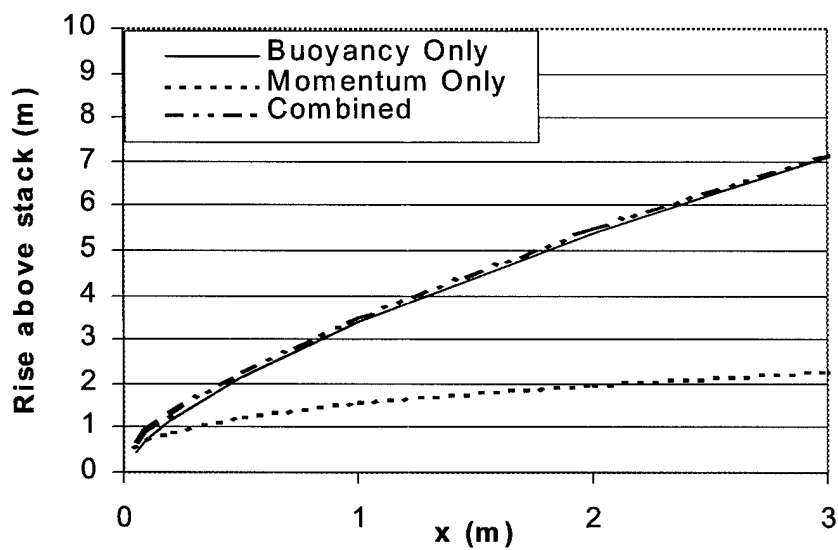


Figure 2.5: Buoyant, Momentum, and Combined Plume Rise for Example Factory Stack

Next, we modify the problem to be more representative of a small roof vent. The problem parameters are

Buoyancy flux parameter ( $F$ )	75 m <sup>4</sup> /s <sup>3</sup>
Stack gas exit temperature ( $T_s$ )	293 K
Ambient temperature ( $T_a$ )	283 K
Wind velocity ( $u_s$ )	4 m/s
Stack gas velocity ( $w$ ) (assumed)	10 m/s
Stack diameter ( $d_s$ ) (assumed)	3 m
Momentum flux parameter ( $F_m$ )	217.3 m <sup>4</sup> /s <sup>2</sup>

The stack effluent is now only slightly warmed over the ambient, which decreases the buoyancy flux parameter and increases the momentum flux parameter (because the density is increased). The stack diameter is a bit large for a roof vent, but is kept constant for comparison.

Figures 2.6 and 2.7 show the rises for this configuration. Because of the temperature difference, the buoyancy rise is about half that for the factory stack, while the total momentum rise is about the same, as the increased momentum parameter is offset by the increased entrainment coefficient. Even with the decreased temperature, buoyancy still dominates the total rise far from the stack. Very close to the stack, however, momentum rise dominates. A bit further from the stack, the difference between the combined rise and the buoyancy rise is roughly constant. It eventually becomes a negligible percentage of the buoyancy rise, but is significant within a few meters of the stack.

These examples may explain the bias of traditional methods toward buoyancy-dominated plumes. Even with a small temperature difference, buoyancy rise is still dominant. With a focus on downstream sampling, little error is made by neglecting momentum rise. Only for an application such as ours, which images the plume very close to the stack, is momentum rise appreciable.

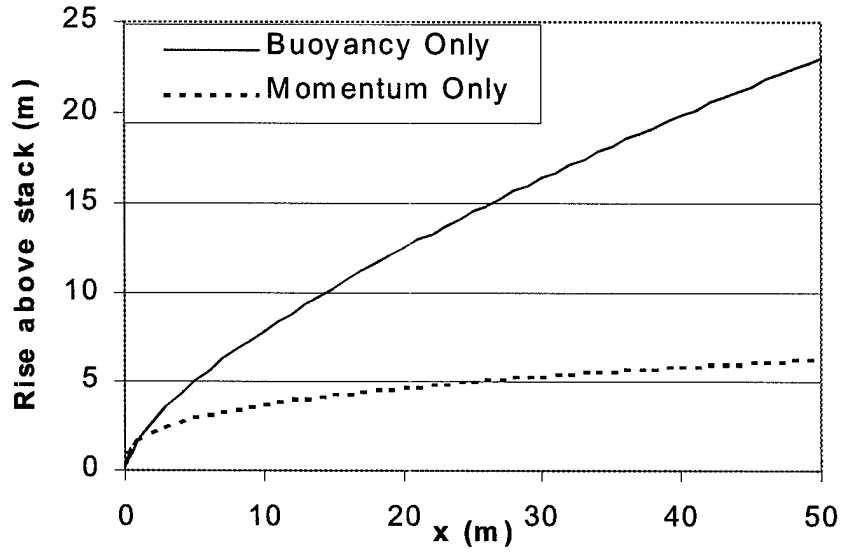


Figure 2.6: Buoyant and Momentum Plume Rise for Example Roof Vent

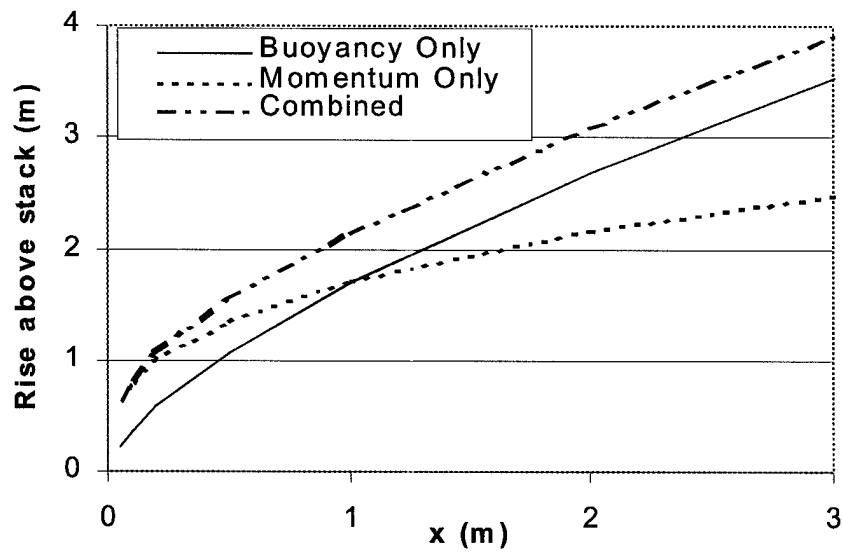


Figure 2.7: Buoyant, Momentum, and Combined Plume Rise for Example Roof Vent

Next, an illustration of the differences in plume rise and dispersion as stability class and terrain type are varied is given. The stack parameters, which are identical for all cases, are

Stack diameter ( $D$ )	4 m
Stack height ( $h$ )	30 m
Wind speed ( $u_s$ )	2 m/s
Stack gas velocity ( $w$ )	5 m/s
Stack gas exit temperature ( $T_s$ )	320 K
Ambient temperature ( $T_a$ )	300 K

Concentration maps for Pasquill stability classes A (unstable), C (neutral) and F (stable) for both rural and urban terrain are shown in the figures of Table 2.1. Higher concentrations are shown as darker shades. All images are in a vertical plane passing through the centerline of the plumes. As expected, the plume disperses more in unstable conditions and less in stable conditions. For a given stability class, it disperses more in urban terrain than rural terrain.

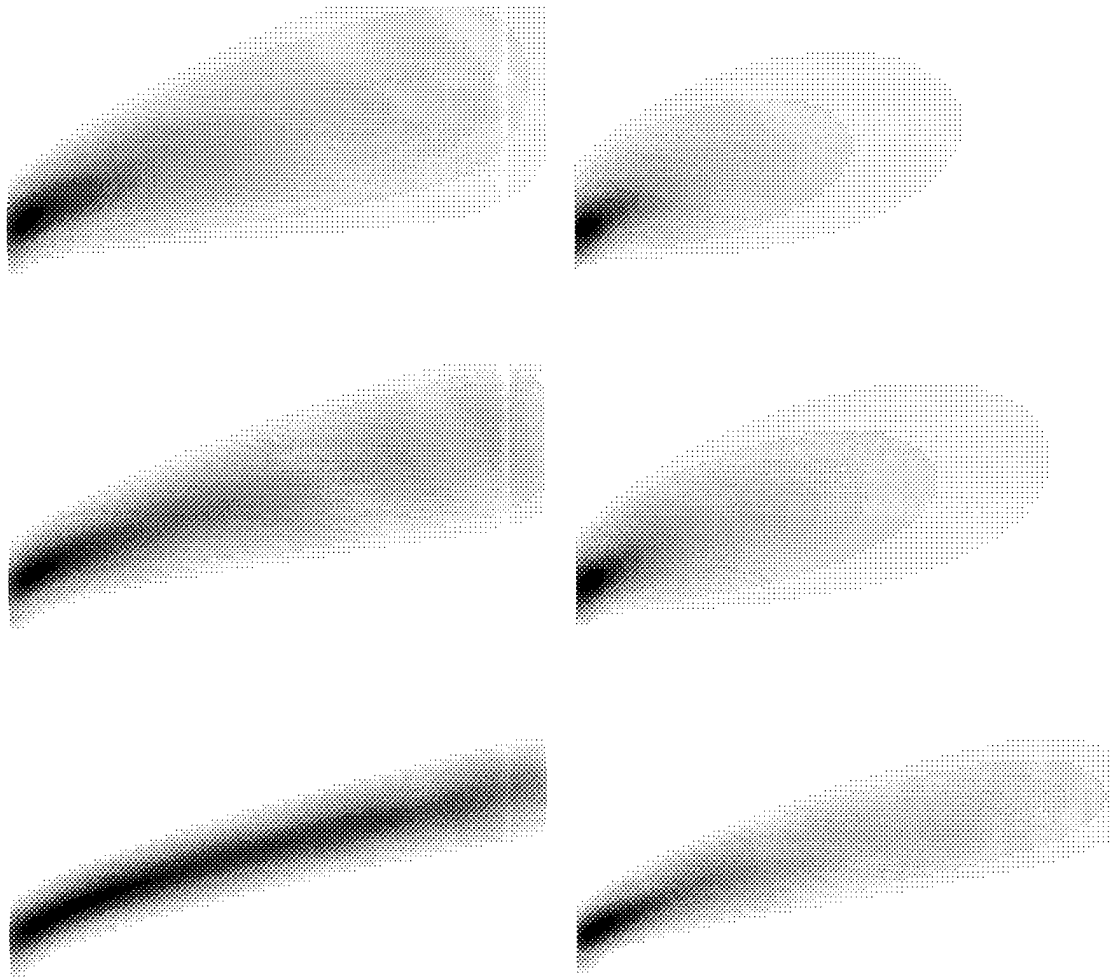


Table 2.1: Plume Rise and Dispersion for Various Conditions. From top to bottom: Stability Classes A, C, F. Left Column: Rural Terrain. Right Column: Urban Terrain.

## 2.2 Advanced Plume Modeling Methods

In the past few years, increases in computer power have enabled far more advanced computer models to be practically used. These techniques more accurately simulate plume behavior in changing meteorological conditions, in complex terrain, and near buildings. Although some of these capabilities can and have been incorporated into traditional models such as ISC3, newer puff models provide a better framework for adding new capability. The basic characteristics of puff models and some advanced capabilities relevant to this research are examined here.

### 2.2.1 Puff Models

The plume models described throughout Section 2.1 are developed for steady-state conditions. Spatial and temporal wind changes are not allowed. Puff models eliminate this restriction by simulating a plume as a series of closely spaced puffs. By tracking puffs individually, more complex behavior can be modeled. The EPA has recently funded the development of the CALPUFF II system. Among other features, this software allows for spatial and temporal wind variations. It includes more advanced modeling of plume behavior in complex terrain than ISC3 does. Other features, such as wet and dry deposition and plume chemical transformation use the same models as ISC3 (Scire et al., 1999).

The main equation for concentration due to a puff in CALPUFF is

$$C = \frac{Q}{2\pi\sigma_x\sigma_y} g \exp\left[-\frac{d_a^2}{2\sigma_x^2}\right] \exp\left[-\frac{d_c^2}{2\sigma_y^2}\right] \quad (2.26)$$

where  $d_a$  and  $d_c$  are the downwind and crosswind distances from the center of the puff,  $\sigma_x$  and  $\sigma_y$  are the standard deviations in the along-wind and cross-wind directions, and  $Q$  is

the pollutant mass in the puff.  $g$  is the vertical contribution, which is given in theory by

$$g = \frac{1}{(2\pi)^{1/2}\sigma_z} \left\{ \exp \left[ -\frac{(H_e - z_r)^2}{2\sigma_z^2} \right] + \exp \left[ -\frac{(H_e + z_r)^2}{2\sigma_z^2} \right] + \sum_{n=1}^{\infty} \left( \exp \left[ -\frac{[2nh - (H_e - z_r)]^2}{2\sigma_z^2} \right] + \exp \left[ -\frac{[2nh - (H_e + z_r)]^2}{2\sigma_z^2} \right] + \exp \left[ -\frac{[2nh + (H_e - z_r)]^2}{2\sigma_z^2} \right] + \exp \left[ -\frac{[2nh + (H_e + z_r)]^2}{2\sigma_z^2} \right] \right) \right\} \quad (2.27)$$

where  $H_e$  is the height of the puff center above ground,  $\sigma_z$  is the standard deviation of the distribution in the vertical direction,  $z_r$  is the height of the point of interest above the ground, and  $h$  is the height of the well-mixed layer that the plume is trapped in (which may be made very large if the atmosphere is unstable). The form of the equation shown here is actually adapted from another source; the form in the CALPUFF manual is valid only at ground level. In these equations, the  $r$  subscripts stand for “receptor.” This term is used throughout the ISC3 and CALPUFF documentation because the codes are traditionally used to simulate the concentrations that would be measured by sampling devices at discrete points, and concentrations in these codes are calculated only at receptors. Since our application does not involve physical receptors, we use the term “point of interest” when possible, but occasionally refer to receptors for consistency with ISC3 or CALPUFF documentation.

The two terms on the first line of Equation 2.27 are the contributions from the real plume and its image below the ground (as in Equation 2.3); this situation is shown in Figure 2.8. Using a virtual source in this manner simulates total reflection from the ground. At a point of interest, the concentration is the sum of the concentration from the actual plume and the concentration from its mirror image.

Extending the use of images to simulate reflection or trapping of the plume below an inversion or mixing lid, additional sets of images can be constructed. These images, and their images in turn, give rise to the summation in Equation 2.27. In the full form, the summation describes an infinite number of reflections between the ground and the mixing

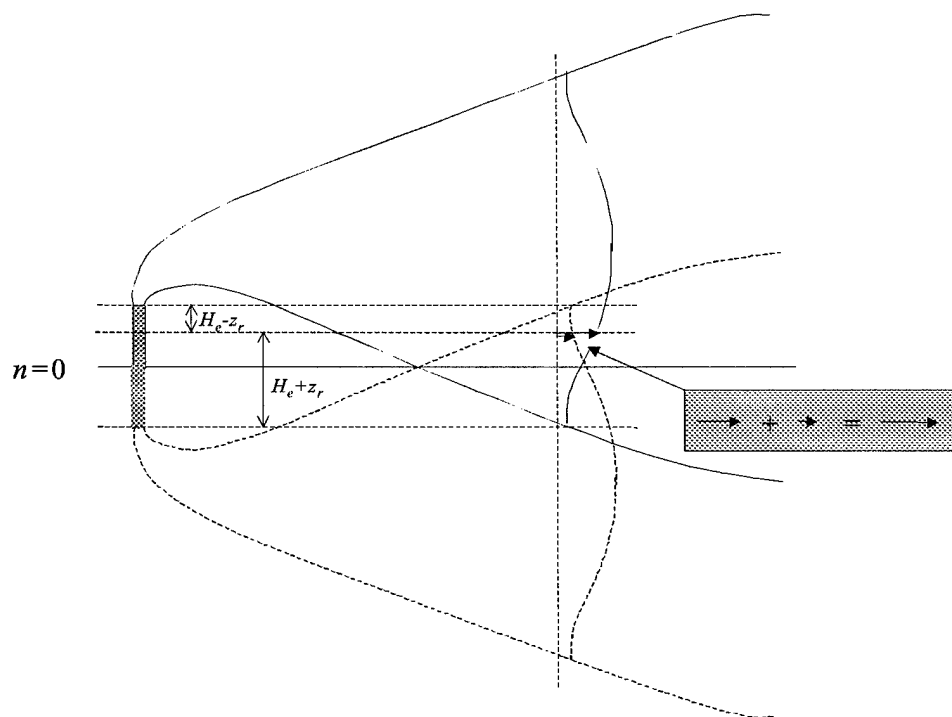


Figure 2.8: Use of Image Stack to Simulate Plume Reflection from Ground

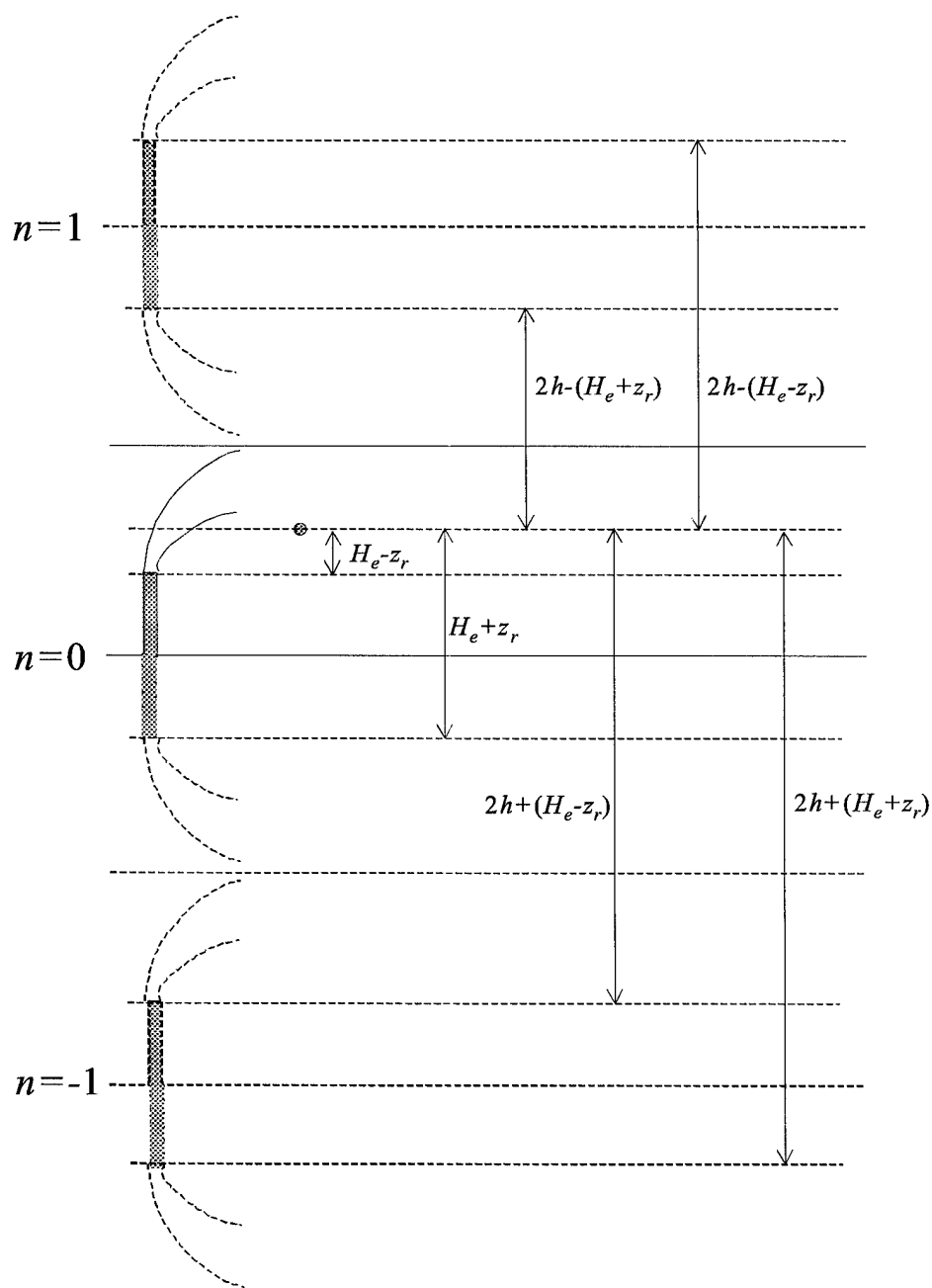


Figure 2.9: Use of Multiple Images to Simulate Plume Reflection from Ground and Mixing Lid

lid.

For the special case of  $n_{max} = 1$ , Equation 2.27 reduces to

$$g = \frac{1}{(2\pi)^{1/2}\sigma_z} \left\{ \exp \left[ -\frac{(H_e - z_r)^2}{2\sigma_z^2} \right] + \exp \left[ -\frac{(H_e + z_r)^2}{2\sigma_z^2} \right] + \right. \\ \exp \left[ -\frac{[2h - (H_e - z_r)]^2}{2\sigma_z^2} \right] + \exp \left[ -\frac{[2h - (H_e + z_r)]^2}{2\sigma_z^2} \right] + \quad (2.28) \\ \left. \exp \left[ -\frac{[2h + (H_e - z_r)]^2}{2\sigma_z^2} \right] + \exp \left[ -\frac{[2h + (H_e + z_r)]^2}{2\sigma_z^2} \right] \right\}$$

which is the form actually used by CALPUFF. The stack images for Equation 2.28 are shown in Figure 2.9. In this figure, the plumes and Gaussian distributions are deleted for clarity. A point of interest is shown, and the manner in which different terms in the equation correspond to different images is illustrated.

The two terms on the second line of Equation 2.28 are the contributions of two imaginary stacks obtained by reflecting the real stack and its original below-ground image about the ceiling of the mixed layer; these are at the top of Figure 2.9. They provide for a single reflection from the mixing ceiling, either directly or after reflection from the ground. The two terms on the third line of Equation 2.28 are obtained by reflecting the two stacks from the second line about the ground; they are at the bottom of Figure 2.9. They thus provide for a bounce from the ground after reflection from the ceiling (allowing for the reflection sequences ground-ceiling-ground and ceiling-ground). Additional terms in the summation would add more images of the source, providing for more multiple reflections of the plume. At some distance downwind, the material becomes evenly distributed between the ground and the mixing ceiling.

It is noted that there is no causal factor in Equations 2.27 and 2.28. No check is made to determine whether the sensible plume has in fact reached either the ground or mixing lid, so that reflection has occurred. Thus, even a fraction of a meter from the stack, the concentration predicted by the equation incorporates all of the bounce terms included in

the series. In practice this introduces minimal error, as a point near the stack is many standard deviations away from any of its images, so that they add only a negligible amount to the total concentration. This non-causality drawback is probably no more severe than the Gaussian assumption itself, which predicts non-zero concentrations at all points in space at any time.

### 2.2.2 Slug Models

Although splitting a plume into puffs offers many advantages over traditional monolithic models, tracking many puffs is computationally expensive. This is a drawback particularly close to a source, where many puffs are needed to accurately represent the curvature of the plume. In order to accurately represent the concentration, puffs should be separated by no more than  $2.0\sigma_y$ , which yields a concentration half-way between the puff centers of approximately 96% of the correct value (Scire et al., 1999). As puffs move downstream, they grow, and thus become more closely spaced when normalized by their dispersion. The computational burden of tracking these puffs may become excessive.

The CALPUFF system introduces a revised puff formulation, called a slug, for these situations. The slug appears as an elongated puff and is equivalent to a series of puffs, each with infinitesimal mass. The formulation for the slug is

$$C(t) = \frac{Fq}{(2\pi)^{1/2}u'\sigma_y} g \exp \left[ -\frac{d_c^2}{2\sigma_y^2} \frac{u^2}{u'^2} \right] \quad (2.29)$$

where  $u$  is the mean wind speed,  $u'$  is the so-called “scalar wind speed” (defined as  $u' = (u^2 + \sigma_v^2)^{1/2}$ , where  $\sigma_v$  is the wind speed standard deviation),  $F$  is a “causality” function, and  $g$  is the same vertical factor as defined in Equation 2.28.  $q$  is the source emission rate, in units of mass per time (as opposed to the total mass for a puff,  $Q$ ).  $d_c$  is the distance from the receptor to the slug, as measured perpendicular to the slug axis. The ratio of wind speeds is included to reasonably handle calm situations; for wind speeds more than a few meters per second, it is effectively unity. Conceptually,  $\sigma_y$  varies along the length of

the slug according to the Pasquill-Gifford curves or other relationships, as it does in the monolithic model; in practice, CALPUFF calculates concentrations at the ends of the slugs and at receptors (points of interest); therefore, a ray tracing code would be responsible for interpolating between the dispersion coefficients at the ends of the slugs as needed.

Neglecting the  $F$  factor, which will be discussed momentarily, the basic form of the Equation 2.29 is a broadening quasi-cylinder with Gaussian decay. Horizontal decay perpendicular to the slug axis is provided directly by the exponential term, while vertical decay is included in the  $g$  term.

The equation as presented so far is essentially identical to the traditional monolithic model of Equation 2.3. The differences are inclusion of the mixing lid effect in the vertical ( $g$ ) term and the use of the scalar wind speed rather than the average wind speed. Neither of these is intrinsic to the slug model; in fact, the mixing lid effect is incorporated in the ISC series of codes. The difference between the slug and the traditional plume model is in the  $F$  factor. This factor limits the spatial extent of the slug in the along-axis direction. It is given by

$$F = \frac{1}{2} \left\{ \operatorname{erf} \left[ \frac{d_{a2}}{\sqrt{2}\sigma_{y2}} \right] - \operatorname{erf} \left[ \frac{-d_{a1}}{\sqrt{2}\sigma_{y1}} \right] \right\} \quad (2.30)$$

where  $d_{a1}$  and  $d_{a2}$  are the components of the distances from the receptor to the “young” (most recently emitted) and “old” ends of the slug in the along-slug direction, with the signs of both chosen so that the positive direction is in the direction of the opposite end of the slug (i. e., the positive directions are opposite, and both point inside the slug).  $\sigma_{y1}$  and  $\sigma_{y2}$  are the horizontal dispersion coefficients at each end of the slug, which are in general different. erf is the “error function,” defined as  $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$  (i.e., it is obtained by integrating the normal function).

An example causality function is shown in Figure 2.10. The end points of the slug are at 20 and 80 units, with dispersion  $\sigma_{y1} = \sigma_{y2} = 2$  units.

Intuitively, this causality function, which could be recast in terms of integrals of Gaussians, works well. Because of the chosen signs on the distances, each of the erfs is positive

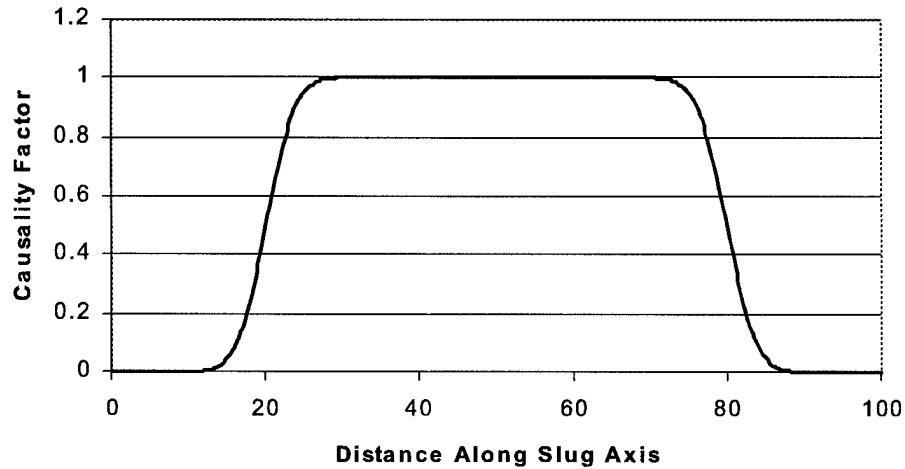


Figure 2.10: Example Causality Function

if its slug end has not reached a point of interest and negative if it has. At points well outside the slug, neither end has passed or both ends have and both erfs evaluate to the same value (either  $+1$  or  $-1$ ); thus, there is no net concentration when the erfs are subtracted. For points well within the slug, the erf for the older end evaluates to  $1$ , and the erf for the younger end to  $-1$ , so that  $F = 1$ . For points near the ends, the erfs vary, and the concentration falls off. It is stated in the CALPUFF manual (Scire et al., 1999) that the “tails” decay in such a way that, as adjacent slugs overlap, the steady-state monolithic solution is produced.

Slugs have not been used in our present work nor implemented in computer code. First, simulation of high-resolution plume meander is much more straightforward with puffs, since slugs cannot be easily bent in the middle. This difficulty is also significant near the stack as the plume is bending over, though CALPUFF should choose to use puffs in this region. We present the theory for the benefit of lower-resolution applications, which may be able to use a coarser slug-based plume, and as a bridge between monolithic techniques and puff-based

techniques.

### 2.2.3 Advanced Dispersion Modeling

The CALPUFF system models a variety of phenomena that enhance the accuracy of any model. Some of these features will not be used in this research, as they conflict with the methods used to introduce meander into the plume trajectory, but they still bear mention. The features include development of conventional dispersion coefficients from actual turbulence data or micrometeorological data, the splitting of puffs that encounter velocity gradients, penetration of the top of the mixed layer, and vertical transport of puffs that converge horizontally. Since none of these factors directly impact the generation of synthetic images of plumes, they will not be discussed in great detail here. The most significant of them are briefly discussed to highlight the capabilities that increase the realism of the modeled plumes.

Approximation of dispersion coefficients based on local micrometeorological data is a significant improvement for regulatory applications. The range of local information that can be used varies greatly. If turbulence measurements are available, they can be used to estimate dispersion. If they are not, statistics on the standard deviation of the wind speed can be used, certainly with no less accuracy than what would be obtained using Pasquill's or other generic values. In a slightly different way, the MIND model (to be discussed subsequently) generates instantaneous diffusion coefficients based on wind data.

CALPUFF also interpolates wind speed based on observations from various stations to generate a spatially varying wind field. In general, this feature would make plume images more visually realistic. For relatively constant winds, average ground concentrations are not likely to change greatly with small changes in wind; however, small changes tend to make plumes look less artificial, a benefit when generating synthetic images. The variable wind speed can also be used as a tool in simulating other conditions. For example, wind speed variation around buildings could possibly be simulated, although this would likely

over-extend CALPUFF's method. This could be used to affect the temperature of the ground and other nearby objects, resulting in more realistic contrast variation in thermal imagery, even in regions where plumes are not present.

#### 2.2.4 Short-Term Models

As has been mentioned, nearly all current models report concentrations averaged over a time on the order of many minutes. This averaging is suitable for many pollutants, where long-term exposure is important. Since these averaging models are computationally simple and most important for environmental applications, they have been developed thoroughly in the past fifty years.

Nearly all imaging devices, on the other hand, capture an instantaneous image of the scene. Thus, for SIG applications, we need to make some correction for the averaging time used in the models. Simplistically, this can be done by reducing the dispersion coefficients to reflect a shorter time period; this is an option in the ISC models. This approach is valid for small changes in averaging time, but not realistic for times less than a few minutes. To attempt to model an instantaneous plume, the fluctuations in the location of the plume centerline, caused by short-term wind fluctuations, must also be characterized.

Fluctuating models are not new; a statistical framework for studying fluctuations was developed in Gifford (1959). The early studies were aimed at comparing peak instantaneous fluctuations to peak long-term fluctuations, so that quick-fix multipliers could be applied to long term models to estimate peak concentrations.

Fortunately, with the increase of computer power in the last 15 years, more advanced short-term models that actually track the location of the plume over time are being developed. The intended use of these models is modeling of highly toxic pollutants, where peak exposure is more important than average exposure. With some adaptation, they are highly suitable to the SIG process.

### 2.2.5 The MIND model

The most significant work found to date in this area is the work of Peterson, Lamb, and O'Neill, of Montana Tech and Washington State University. Through the early 1990's, these researchers collected high frequency (1 Hz) wind data and developed techniques and models that give results similar in character to the empirical data (Peterson, Lamb, and Stock, 1990), (Peterson, Ballard, and Lamb, 1995), (Peterson and Lamb, 1995, 1996), (Mazzone and Peterson, 1996), (Peterson, O'Neill, and Lamb, 1996), (O'Neill, 1996a,b), (Peterson, Mazzolini, O'Neill, and Lamb, 1999).

The end product is a model known as the Meandering INSTantaneous Diffusion (MIND) Model (O'Neill, 1996b). This program takes high-frequency wind data (assumed to be taken at the plume source) and general information about the conditions (Pasquill-Gifford stability, etc.) and calculates the progress of a plume over time. Two features of this model are important to the current research.

#### Wind Smoothing

First, the high-frequency wind data are smoothed to reduce the random effects of turbulence away from the location where the data were taken. This is done by estimating the travel time from the plume source to a given location according to

$$t = \frac{r}{(U + 2.15\sigma_U)} \quad (2.31)$$

where  $r$  is the distance from the plume source to the point of interest,  $U$  is the mean wind speed, and  $\sigma_U$  is the standard deviation of the wind speed. This travel time is then used as a window for a moving average process. Thus, for points of interest close to the source, the travel time is short, the data are not smoothed very much, and random variations remain. Further from the source, a wider window is used, the wind is smoothed more, and the average wind speed dominates plume behavior. The wind speed used in calculating the smoothing window is the 90th percentile speed (i. e., the mean plus 2.15 standard deviations), which

makes the travel time shorter than it would be using just the mean wind speed, preserving more of the random character (Peterson et al., 1990).

### Instantaneous Diffusion

Second, the MIND model finds instantaneous diffusion coefficients through one of several equations developed and tested by the researchers; these depend on the long-term diffusion (from the traditional Pasquill-Gifford relations) or on standard deviations of wind angles. For example, the default instantaneous dispersion coefficient is

$$\sigma_{yi} = \sigma_{zi} = 0.2849\sigma_{\theta R}r \quad (2.32)$$

where  $\sigma_{\theta R}$  is obtained by taking the standard deviation of the wind direction over each smoothing window and averaging the results and  $r$  is the distance from the source. In a more recent paper (Peterson et al., 1999), this relationship is revised to

$$\sigma_i = (\sigma_{yi}\sigma_{zi})^{0.5} = 0.2849\sigma_{\theta R}r \quad (2.33)$$

Although no guidance is given for choosing the individual coefficients, this formula gives some method for varying the coefficients with respect to each other. One possibility, other than setting the two coefficients to equal each other, is to set the two coefficients so that the ratio of the vertical coefficient to the horizontal coefficient is the same as the ratio for the one-hour Pasquill-Gifford coefficients. Another possibility would be to choose the coefficients to simulate a particular plume phenomenon such as a coning plume, which disperses more in the vertical direction, or a fanning plume, which disperses more horizontally.

### Concentration Calculation

Following calculation of the smoothed wind speed and the instantaneous dispersion coefficients, the concentration at a point of interest can be calculated using a Gaussian equation. This portion of the MIND code is not actually used in the current research; rather, only the smoothed wind direction is actually found by the original MIND code.

It is assumed, as in the MIND code, that the centerline is located in the direction of the smoothed wind speed from the source. (Because of this assumption, MIND does not actually propagate material in time; the plume at any instant is effectively made up of segments of non-meandering plumes that each originate at different points and point in slightly different directions.) MIND's estimates of the instantaneous diffusion are used, as they are much better suited to the present task than CALPUFF's estimates.

### 2.2.6 Interaction of CALPUFF with MIND

Unfortunately, some features of CALPUFF are incompatible with the use of the entire MIND program. While CALPUFF models terrain effects, MIND operates under the assumption of a flat earth. In the case of puff location, we reconcile the two by using the plume centerline shape output by MIND as a perturbation of the puff locations found by CALPUFF, as shown in Figure 2.11. This seems a reasonable approach unless terrain is too severe; even then, the perturbed plume will likely look "about right."

Because we treat the MIND-derived wind field as a perturbation and apply it to the CALPUFF plume regardless of its orientation, we should not use the exact same random wind sequence for plumes within the same scene. If we were to do so, fluctuations calculated by MIND would affect all plumes, as shown in Figure 2.12. In particular, large motions would be visibly seen in all plumes simultaneously. To avoid this, multiple sequences with the same statistics are used.

The dispersion calculations of the two models, however, are not compatible. Thus, CALPUFF's modeling of building-induced dispersion, which locally increases the dispersion coefficients, and CALPUFF's variation of dispersion coefficients with terrain type, cannot be easily reconciled with MIND's dispersion coefficient calculations, which are based solely on wind statistics. As MIND's dispersion relations are based on short-term data, it is more reasonable to use MIND's calculations. Even if it were possible to incorporate CALPUFF's dispersion effects, this might not be wise, as the building-induced dispersion and other

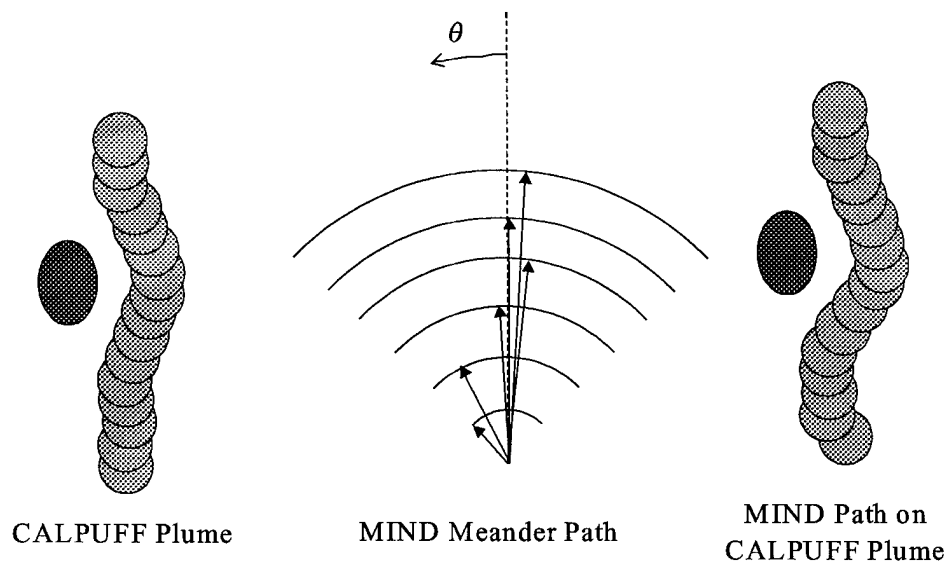


Figure 2.11: Superposition of CALPUFF and MIND Calculations

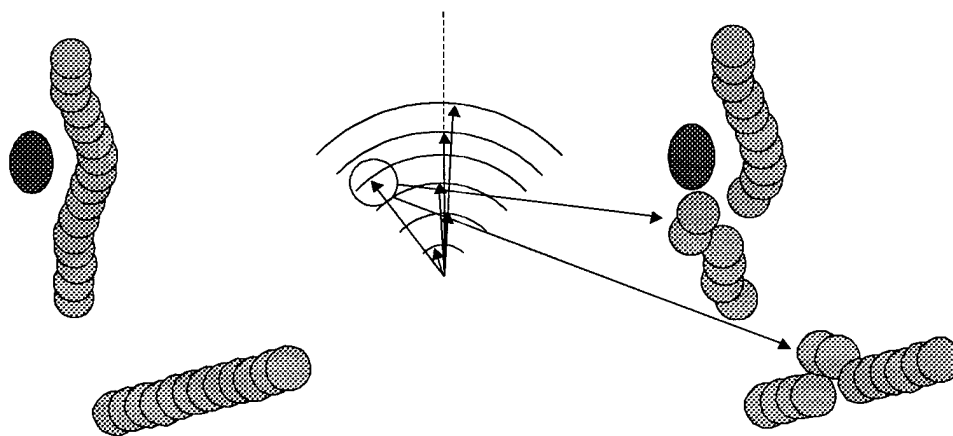


Figure 2.12: Effect of Wind Perturbations on Multiple Plumes

effects are based on curve-fits to one-hour average data, and thus might not be scaleable with any reliability to shorter time periods. In summary, our composite CALPUFF-MIND plume is deflected by terrain (since the baseline position is determined by CALPUFF, which models variable wind flow), but the dispersion is not affected by terrain or land use type.

## 2.3 Wind Modeling

Using the techniques of the MIND model just discussed to simulate the instantaneous position of a plume requires high-frequency wind data (which is then smoothed). Although it would be possible to use experimental data, a more attractive approach is to derive a model for the wind process and generate random wind fields for each scene. This approach would make each plume unique, and might also contribute some information about how the wind parameters vary with changing stability class. The goal of this section is to develop a model that is able to generate artificial wind sequences that are statistically similar to actual wind data. In statistical terms, the wind speed and direction are time series.

The problems of modeling and forecasting time series data have been studied extensively in the statistics community over the past thirty years. The principal work cited as the foundation in this area is Box and Jenkins (1976). The procedures for developing a Box-Jenkins Model, also known as an ARIMA (AutoRegressive Integrated Moving Average) model for a particular series have also become well established (see Bowerman and O'Connell (1987)).

Past studies have developed ARIMA models for wind patterns over long time periods (years) (Breckling, 1989). Also, Hanna (1979) has shown that, when averaging over a significant number of similar data points, the wind speed at a particular time is linear with the speed at a previous time (which will be described shortly as an autoregressive model of order 1). These results suggest that investigating the use of an ARIMA model for high-frequency wind data is appropriate.

### 2.3.1 Taxonomy and Definitions of ARIMA Models

The general ARIMA model consists of three separate types of behavior, which can be combined in various ways as shown in the table below.

Type of Model	Nomenclature
Autoregressive	AR
Moving Average	MA
Autoregressive Moving Average	ARMA
Integrated Moving Average	IMA
Autoregressive Integrated Moving Average	ARIMA

#### AR Models

The first type of model, the autoregressive process, is governed by the behavior

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + \cdots + \phi_p \tilde{z}_{t-p} + a_t \quad (2.34)$$

where  $\tilde{z}_t = z_t - \mu$  is the deviation from a fixed value of the newest value in the time series,  $\tilde{z}_{t-1}, \dots, \tilde{z}_{t-p}$  are the  $p$  most recent deviations in the time series,  $\phi_1, \dots, \phi_p$  are constant weights, and  $a_t$  is a white noise random variable, known as a “shock,” with zero mean and variance  $\sigma_a^2$ . The value  $\mu$  is the true mean if the process is statistically stationary<sup>1</sup>; if it is not, it is some other arbitrary level. This model is known as an autoregressive model of order  $p$ , or an AR( $p$ ) model. As can be seen from the equation, the deviation from the mean for the present value is a weighted sum of the deviations for the  $p$  immediate past values, with a single new random element.

#### MA Models

The second process is the moving average process. Its governing equation is

$$\tilde{z}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q} \quad (2.35)$$

---

<sup>1</sup>A process is said to be stationary if certain statistical properties are invariant. Practically, we are concerned with whether the mean wind direction for one arbitrary group of data points is the same as the mean for all other arbitrary groups. This, of course, is not strictly true of the wind over long time periods, but may be true for limited periods of interest.

where  $\theta_1, \dots, \theta_q$  are constant weights, and  $a_t, \dots, a_{t-q}$  are a series of shocks. This model is known as a moving average model of order  $q$ , or a  $MA(q)$  model. In this type of process, the present value is not directly a function of the immediate past values. The values will, however, tend to be correlated (or perhaps negatively correlated) for small lags since the value  $z_t$  and the value  $z_{t-n}$  ( $0 \leq n \leq q$ ) are both functions of the set of shocks  $\{a_{t-n}, \dots, a_{t-q}\}$ .

### ARMA Models

Taken together, the autoregressive and moving average models can be combined into an  $ARMA(p, q)$  model. This model is probably the most commonly used time series model. The combined governing equation is

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + \dots + \phi_p \tilde{z}_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q} \quad (2.36)$$

### IMA and ARIMA Models

The ARMA model, along with its constituents, the AR and MA models, are suitable for processes that are inherently statistically stationary, so that the mean is constant over short periods of time. Many processes do not meet this criterion, but seem to have a gradually fluctuating behavior about a long-term mean average. For these processes, a sequence of differences between values at a certain interval tends to be stationary. The integrated moving average (IMA) and autoregressive integrated moving average (ARIMA) models account for this behavior. Conceptually, these models are found by replacing every occurrence of  $\tilde{z}_{t-n}$ ,  $n = 0, 1, \dots, p$  in either an MA model or an ARMA model with some form of difference  $z_{t-n} - z_{(t-n)-x}$ .

Two types of differencing are used. The nonseasonal operator replaces  $\tilde{z}$  with  $\nabla^d z_t$ , the “ $d$ th regular difference.” The differences are given by  $\nabla z_t = z_t - z_{t-1}$ ,  $\nabla^2 z_t = (z_t - z_{t-1}) - (z_{t-1} - z_{t-2}) = z_t - 2z_{t-1} + z_{t-2}$ , etc. The first difference is sufficient for processes that change in level, but have approximately zero slope over any region of interest. The second difference allows for fluctuations in both level and slope (Box and Jenkins, 1976).

In operator notation,  $\nabla^d z_t = (1 - B)^d z_t$  where  $B^n z_t = z_{t-n}$  (It is noted that  $\nabla z_t = \nabla \tilde{z}_t$ .) This nonseasonal operator accounts for miscellaneous fluctuations.

A second type of differencing operator, known as a seasonal variation, is of the type  $\nabla_L = 1 - B^L$ , where  $L$  is the appropriate number of “seasons” (quarters, months, weeks, etc.). This operator is used to account for data that experience regular temporal changes, such as resort occupancy, retail sales, etc. (Bowerman and O’Connell, 1987). This operator can also be raised to some power, as  $\nabla_L^D = (1 - B^L)^D$ .

Differencing operations can be added to the AR and MA models in two ways. Traditionally, the data are pre-analyzed to determine whether seasonal or nonseasonal differencing, are necessary. At the same time, other transformations (e. g., logarithmic transformations) can be made. This approach is assumed in Bowerman and O’Connell (1987), which denotes the original data as a series  $y_t$ , the data after a logarithmic or other similar transformation as  $y_t^*$ , and the final data after a “general stationarity transform” as  $z_t = \nabla_L^D \nabla^d y_t^* = (1 - B^L)^D (1 - B)^d y_t^*$ , which is then fed in to the ordinary time series model as stationary data. Modern software can apply the differencing operation at the same time as the other operators, as will be discussed shortly.

### Constant Term

The previous forms of the models have all assumed that the process in question, as submitted to the AR, MA, or ARMA models, is stochastic, i. e., that the true mean of all realizations of the process is uniform. For non-stationary data, a general stationarity transform was discussed. If the only non-stationary trend is a constant drift of the local average, however, the full differencing transformation can be avoided through the addition of a constant term to the model. In these cases, the  $\mu$  embedded in the  $\tilde{z}$ s contained in the models is some convenient level, such as the estimated value of the true mean as determined by the average of a set of experimental values, rather than an actual true mean.

The trend is added to any of the models by simply adding a constant to the right hand

side of any of the models. This term is denoted as  $\theta_0$  in Box and Jenkins (1976) (notionally lumping it into the MA operator) and  $\delta$  in Bowerman and O'Connell (1987). As a result of this term, each value in any sort of model will tend to be shifted a small amount in one direction from the value predicted by a stochastic model, and the predicted time series will wander over time. A constant is not necessary if the data have already been made stationary through differencing operations or other transforms (alternatively, it could be included in the analysis; if it turns out to be statistically significant, then the transformed data are not stationary, indicating a problem with the original transformation).

The constant term can also be used for convenience to eliminate the need to subtract the mean from each data point. For example, the autoregressive equation (Equation 2.34) can be rearranged as

$$\begin{aligned} z_t - \mu &= \phi_1(z_{t-1} - \mu) + \phi_2(z_{t-2} - \mu) + \cdots + \phi_p(z_{t-p} - \mu) + a_t \\ &= (-\phi_1 - \phi_2 - \cdots - \phi_p)\mu + \phi_1 z_{t-1} + \phi_2 z_{t-2} + \cdots + \phi_p z_{t-p} + a_t \end{aligned} \quad (2.37)$$

so that

$$z_t = (1 - \phi_1 - \phi_2 - \cdots - \phi_p)\mu + \phi_1 z_{t-1} + \phi_2 z_{t-2} + \cdots + \phi_p z_{t-p} + a_t \quad (2.38)$$

Using the constant in this manner is convenient, but also hides immediate evaluation of whether there is a non-stationarity being corrected by the constant.

### **Parsimonious Models**

Following traditional methods, raw data are first transformed to make them stationary, and the transformed data are then fed to the appropriate AR, MA, or ARMA model. With more modern computer software, the differencing process can be done at the same time as the ARMA modeling. This is much more convenient, but it possibly encourages inclusion of terms beyond what are necessary for a parsimonious model (i. e., the model with the fewest terms of those that adequately describe the data). Thus, when running a full ARIMA

model, the user should first examine plots of the data with and without differencing to determine whether it is necessary, and should examine the error of the model and the statistical significance of new terms as the order of the AR and MA processes are increased.

### 2.3.2 Theoretical Behavior of Models

#### Measures of Behavior

The tentative identification of an appropriate time-series model for a given set of data (as summarized in Bowerman and O'Connell (1987)) is made by examining the Sample Autocorrelation Function (ACF) and Sample Partial Autocorrelation Function (PACF) and comparing these to the general behavior of the theoretical ACF and PACF for a given model. (The ACF and PACF are referred to as the SAC and SPAC by Bowerman and O'Connell (1987)).

#### Sample Autocorrelation Function

Fundamentally, the sample autocorrelation function at lag  $k$  of the series  $z_b, z_{b+1}, \dots, z_n$  is the expected value of  $\tilde{z}_{t-k}\tilde{z}_t$ , which can be calculated by

$$r_k = \frac{\sum_{t=b}^{n-k} (z_t - \bar{z})(z_{t-k} - \bar{z})}{\sum_{t=b}^n (z_t - \bar{z})^2} \quad (2.39)$$

This quantity measures the tendency of data points spaced by  $k$  time units ("lag  $k$ ") to move together in a linear fashion. It is easily seen that  $r_0 = 1$ , and it can be proven that  $-1 \leq r_k \leq 1$ . For an AR process, the ACF is infinite in extent since each value is a weighted function of the previous values. The function gradually decays, but there is always a residual chain of influence. For an MA process, on the other hand, the ACF is zero at lags greater than the order of the MA process. This is not surprising, as two values at greatly different times are functions of different random shocks.

## Sample Partial Autocorrelation Function

The meaning of the partial autocorrelation is difficult to describe. Intuitively, Bowerman and O'Connell (1987) state, "This quantity may intuitively be thought of as the sample autocorrelation of time series observations separated by a lag of  $k$  time units *with the effects of the intervening observations eliminated.*" The sample partial autocorrelation at lag  $k$ ,  $r_{kk}$ , is derived by finding the highest-order weight,  $\phi_{kk}$ , that would be obtained if an  $AR(k)$  model (as in Equation 2.34) were used to model an actual  $AR(p)$  process. It can be shown (as one would expect) that, for AR processes,  $r_{kk} = 0$  for  $k > p$ . Thus, the partial autocorrelations of an  $AR(p)$  process are a set of  $p$  numbers; through a relationship known as the Yule-Walker equations, they completely describe the (infinite) autocorrelation function (Box and Jenkins, 1976). For an MA process, the PACF is very complicated (even Box and Jenkins (1976) do not provide details), but is in general a sum of two exponentials (either real or imaginary), and is thus a decaying function or a damped sine wave.

The sample partial autocorrelation function at lag  $k$  can be estimated through a regressive relationship developed by Durbin as

$$r_{kk} = \begin{cases} r_1 & \text{if } k = 1 \\ \frac{r_k - \sum_{j=1}^{k-1} r_{k-1,j} r_{k-j}}{1 - \sum_{j=1}^{k-1} r_{k-1,j} r_j} & \text{if } k = 2, 3, \dots \end{cases} \quad (2.40)$$

where

$$r_{kj} = r_{k-1,j} - r_{kk} r_{k-1,k-j} \quad \text{for } j = 1, 2, \dots, k-1$$

Box and Jenkins (1976) point out that this algorithm is very sensitive to rounding errors and should not be used if the process is close to being non-stationary. It is not clear from the documentation whether Minitab, one of the packages used to fit models, uses this method or another; the other model, R (R Development Core Team, 2001), uses the Yule-Walker method.

### 2.3.3 Model Identification

The theoretical characteristics of the ACF and PACF are the primary tool for determining whether a process is AR or MA. Pure AR processes have infinite, decaying ACFs and PACFs that abruptly “cut-off” at lags greater than the order of the process, while pure MA processes have ACFs that cut-off and PACFs that decay. (This duality between the AR and MA processes has other aspects, but the details are not relevant to the task at hand.)

To determine the appropriate type of model to be used, the ACF and PACF for the empirical data in question are estimated using Equations 2.39 and 2.40. Each is then characterized as either “cutting off” (high values or “spikes” at low lags, abruptly changing to low values) or “dying down” (damped exponential or sinusoidal). From these results, a tentative model is chosen as indicated in the following matrix (advice adapted from Bowerman and O’Connell (1987); see that source for more detailed information). We present the model development results for our wind data in Section 3.2.3

Behavior	ACF dies down	ACF cuts off after lag $q$
PACF dies down	ARMA (1-2 order)	MA( $q$ )
PACF cuts off after lag $p$	AR( $p$ )	No theoretical model; try AR(1) or MA(1)

### 2.3.4 Varying Wind Speed Effects

In addition to the cross-wind meander incorporated using the techniques of Peterson et al., we introduce a motion of puffs in the downwind direction due to fluctuations in the wind speed. Overall, the distance is corrected according to the ratio of the time to reach its position as calculated from the average wind speed to the time according to the fluctuating wind speed. Incorporation of the method requires, therefore, a wind speed sequence, which can be either experimental or synthetic. We have chosen to use a synthetic sequence of ten-second averages (consistent with the wind speed) that are Gaussian random variables with the appropriate mean and standard deviation. These are then up-sampled along with

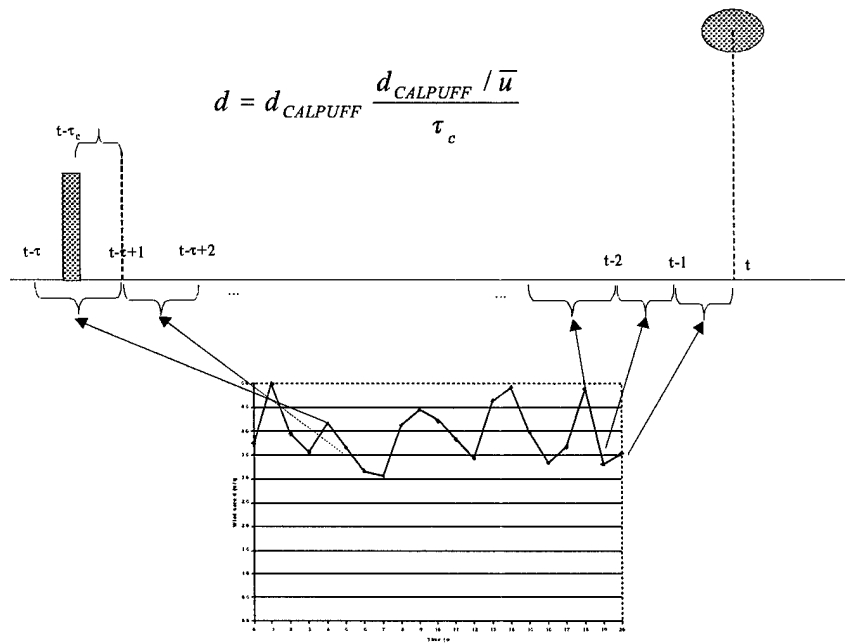


Figure 2.13: Calculation of Distance Correction Factor from Wind Speed Sequence

the direction to produce a high-frequency speed history. This approach is simpler than directly modeling the speed as an autocorrelated sequence at high frequency. Although the speed is autocorrelated at lags of a few seconds it is uncorrelated for ten-second averages. As the effect of the axial puff motion is not as noticeable as crosswind meander, neglecting the autocorrelation is not likely a source of significant error.

The process of calculating the distance correction is shown graphically in Figure 2.13. Beginning at the location of the puff at the time of interest, as given by CALPUFF, we step backward from that point in the wind speed sequence. Each second, the wind speed describes how far the puff travelled in that second. Summing the speeds, we continue until we reach a distance greater than the distance from the stack. The accumulated time to this point is  $\tau$ . We then correct  $\tau$  to account for the fractional second by which the stack was overshoot; the revised travel time is  $\tau_c$ .

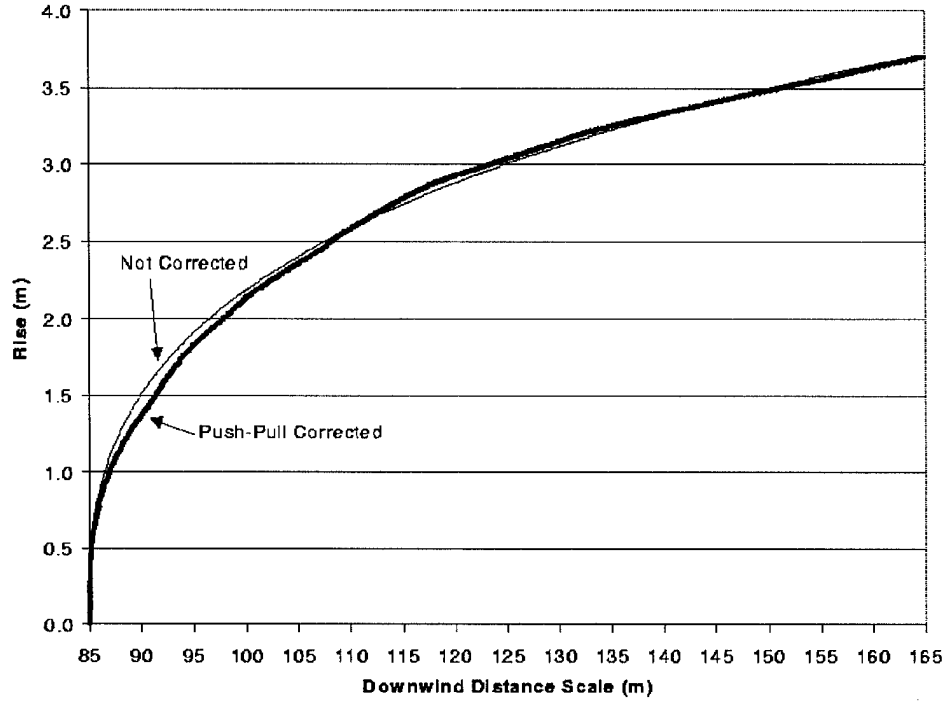


Figure 2.14: Comparison of Plumes With and Without Downwind Puff Movement

As stated above, we divide the travel time derived from the mean wind speed by the travel time from the random wind sequence. This ratio is then multiplied by the distance from the source from CALPUFF to give a new distance:

$$d = d_{CALPUFF} \frac{d_{CALPUFF} / \bar{u}}{\tau_c} \quad (2.41)$$

where  $d$  is the revised distance from the source,  $d_{CALPUFF}$  is the distance from CALPUFF,  $\bar{u}$  is the mean wind speed, and  $\tau_c$  is the corrected travel time.

A comparison of plume rise with the downwind puff movement correction to an uncorrected plume is shown in Figure 2.14.

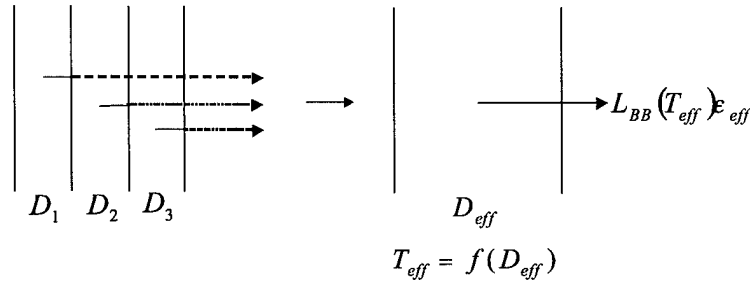


Figure 2.15: Effective Property Method for Calculation of Radiance

## 2.4 Numerical Integration of Concentration

Whatever the method used to find the concentration at a point, the absorption caused by the concentration must be integrated along a path in order to find the total extinction along a ray, which is then used to produce a synthetic image of the plume. This integration could be most easily done if an analytic solution were available; unfortunately, this is not the case.

For monolithic plumes (i. e., those from ISC3 or similar models), integration of concentration is difficult. The plume is not easily identified as an entity; although the concentration can be found at a particular point, the boundaries of the plume in general are not easily described. As a result, past efforts involving the optics of plumes (Kuo, 1997) have relied on average concentrations within a small volume, and the concentration is then assumed to be constant within the volume. This concept is illustrated in Figure 2.15, which shows how a plume consisting of layers with differing properties is collapsed into a single layer with effective properties. This makes the calculation of emitted radiance for a monolithic plume tractable, but introduces error.

Breaking the plume into puffs makes the problem of integration more tractable. It is a reasonably simple matter to define bounds for each puff or slug beyond which the

concentration is negligible; all points that are more than  $n$  standard deviations in any direction from any puff are not in the plume. Similarly, the boundaries of the plume in any desired area can easily be found. Then, the path a ray takes through the region of sensible concentration around a puff or slug can be determined.

Unfortunately, the concentration along this path cannot always be easily integrated analytically. Although the formulation for a puff is a trivariate normal distribution which is well-behaved and seems simple, the particular concentration profile along a path through the puff is not always simple. It is a product of two normal distributions, for the vertical and horizontal directions, which peak at different locations. An example is shown in Figure 2.16. This figure shows the geometry for a sample skewed ray. The peak of the horizontal Gaussian ( $x$  and  $y$ ) occurs at the midpoint of the ray, but it is far from the puff in the  $z$  direction at this point. The peak of the vertical distribution occurs at the very end of the ray. Figure 2.17 shows the normalized concentration through the puff. The peak is not at the very end of the ray, highlighting the interaction between the vertical and horizontal Gaussian terms. The resulting concentration is not Gaussian; Figure 2.17 also shows a Gaussian curve with the same peak location and a standard deviation chosen to roughly fit the right-hand side of the concentration curve. It can be seen that the left-hand side of the concentration curve is higher than the Gaussian.

If all puffs were far from any other objects, it might be feasible to calculate the concentration at a few points, fit a Gaussian to the profile, and integrate the absorbance of the complete puff analytically. However, in many cases, a puff will be close enough to the ground to have substantial reflection. In these cases, the concentration profile is not always even roughly Gaussian. Finally, analytic integration through a slug is probably not tractable. Therefore, we resort to numerical integration. Although computationally burdensome, the approach is more flexible and applicable to a wider range of cases.

The path a ray takes through a puff or slug can be divided into small segments, and the concentration easily found at the segment boundaries. A simple form for the concen-

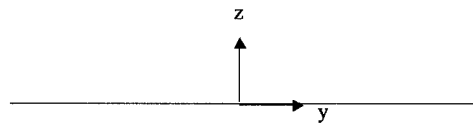
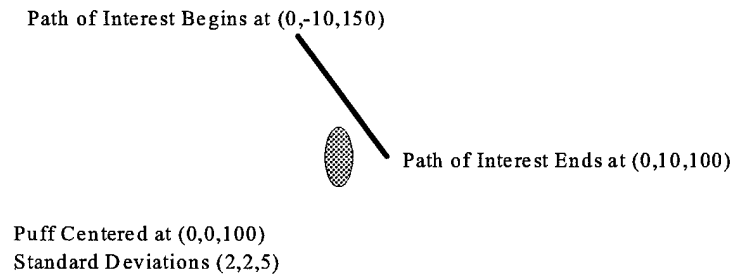


Figure 2.16: Geometry for a Skewed Ray through a Puff

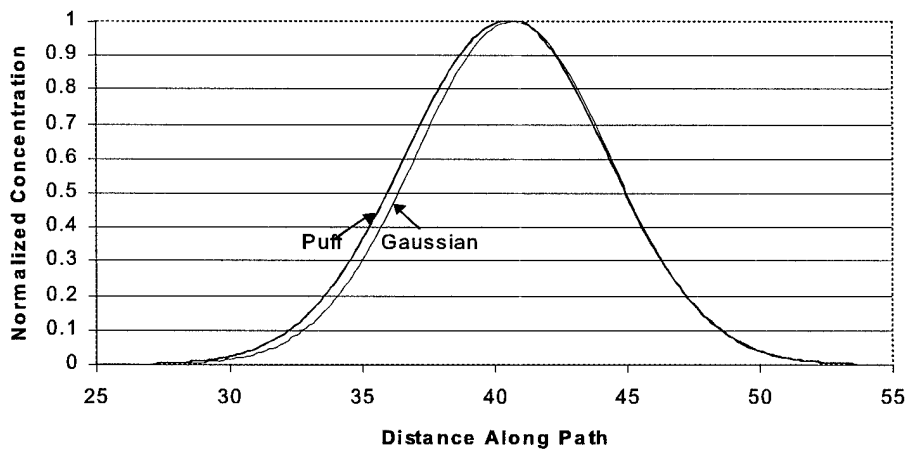


Figure 2.17: Concentration Along the Skewed Ray through a Puff

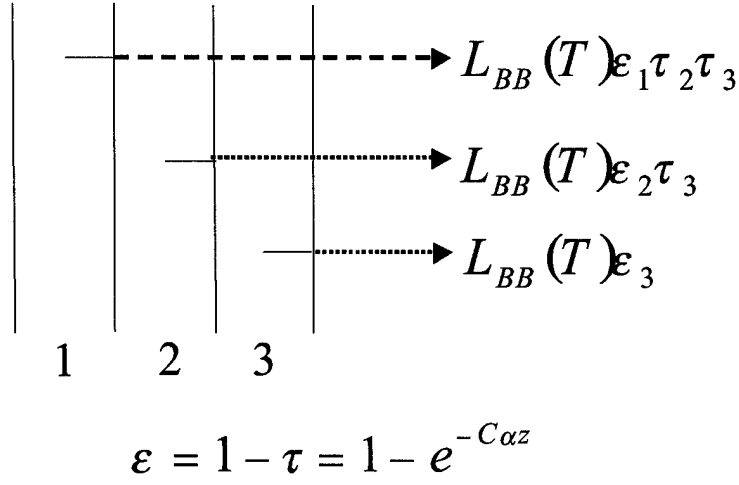


Figure 2.18: Calculation of Thermal Emission

tration between the points (e. g., linear or parabolic) can be assumed, and the integrated concentration along the entire ray or any part of it can then be approximated with a high degree of accuracy. This same method works for both puffs and slugs, and it is also usable for integrating through only a portion of the plume, as is necessary when calculating plume emission or when the target is within the plume.

Self-emission in the plume is calculated by assuming that each step along the path is a homogeneous layer. Then, as shown in Figure 2.18, the emissive contribution of each layer is calculated as the expected radiance for a given temperature and emissivity that is not subsequently absorbed by a later layer. The emissivity of each layer is a function of the particle concentration in the layer ( $C$ ), which we have determined from our model, the absorption cross-section of the chemical species ( $\alpha$ ), which is known from experiment, and the thickness of the layer ( $z$ ), which can be set in our algorithm.

## 2.5 Roof Vent Modeling

As mentioned in the Introduction, a primary motivation for our study of roof vents is their impact on their surroundings. Given a concentration model for a particular case, direct observation of vent gas may be simulated using the same algorithms as used for factory stacks. Our emphasis in this section is on indirect observation, specifically the heating of horizontal plane surfaces such as roofs or the ground by nearby heated discharges. (Although we speak of “roof vents,” our method is applicable to other discharges onto the ground or any other horizontal surface.)

### 2.5.1 Roof Vent Types

We consider three types of vent structures: transverse vents, axial vents, and downward vents. A conceptual diagram of the three types is shown in Figure 2.19.

Transverse vents discharge upwardly and perpendicularly, or nearly so, to the freestream wind. They are thus, in some ways, simply smaller versions of factory stacks, and analysis methods similar to those used for larger stacks are available. Axial vents discharge horizontally in the direction of the freestream wind. Although these vents are likely not as common as other types, the existence of a closed-form solution to the flowfield allows for easy simulation of this type of vent. Downward vents, as the name suggests, discharge downward onto the roof; the jet is deflected by the freestream wind and the roof itself. To our knowledge, no theory satisfactorily models the temperature and flow in such a case. Thus, we resort to a Computational Fluid Dynamics (CFD) solution.

Other vent geometry types are certainly possible; horizontal jets discharging in directions other than the freestream wind are an example. For some cases theoretical solutions may be available, which could be implemented in a manner similar to our axial jet model. Other flows are practically solvable only by CFD; these could be modeled using the techniques we use for downward vents.

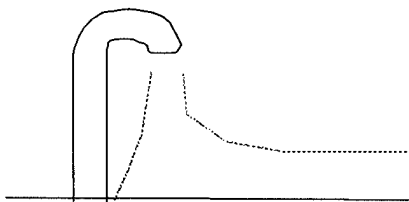
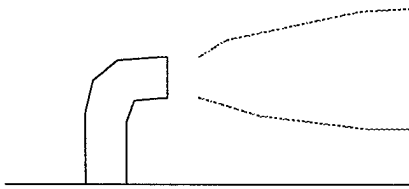
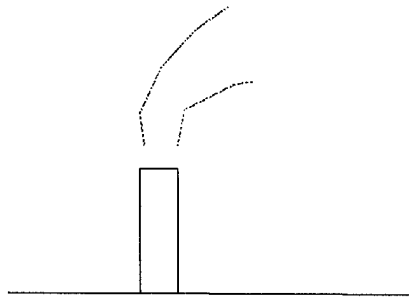


Figure 2.19: Conceptual Illustration of Vent Types: Transverse (top), Axial (middle) and Downward (bottom)

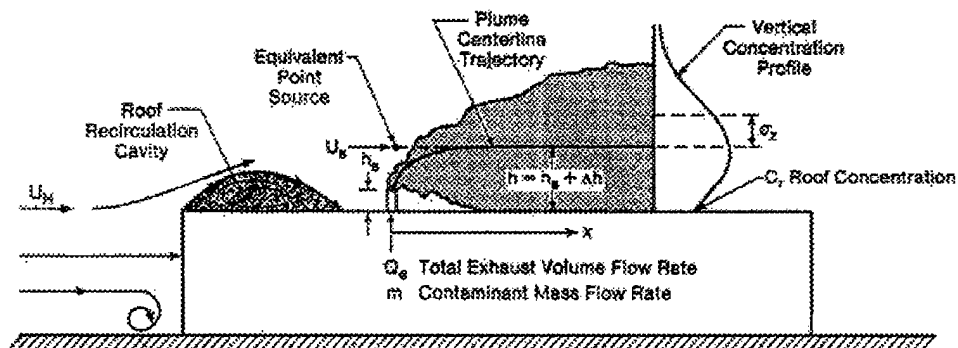


Figure 2.20: Schematic of Roof Vent Flow (from Wilson (1979))

### 2.5.2 Transverse Vent Geometry

The most basic configuration for a roof stack is shown in Figure 2.20. At the leading edge of the building, air rising up over the top creates a recirculation region; air in this region is very turbulent, and any effluents penetrating this area would become uniformly mixed. Estimates of the length of this region have been published (Wilson, 1979). For the most part, the size of this cavity is of secondary importance, as most stacks will be outside of the region in accordance with the recommended practice of the American Society of Heating, Refrigeration, and Air Conditioning Engineers (ASHRAE). With the exception of the recirculation region, the geometry of the factory stack is very similar to that of a stack rising from ground level. As a result, very similar models are applicable. However, some modifications may be appropriate due to nonlinear scaling factors.

### 2.5.3 Transverse Vent Rise Calculations

Rooftop vent plumes have a higher relative level of turbulence than factory plumes. First, because the vent plumes are smaller, the scale of the ambient turbulence is larger relative to the plume than it is for larger factory plumes. Second, factory plumes are usually

emitted from tall, relatively isolated stacks, while rooftop vents are typically not tall enough to completely penetrate the turbulence typically found on roofs, which is due to flow around the building, corners on penthouses, etc. Because of this different environment, changes to the empirical rise equations should be entertained.

One type of modification to the equations is to change the various entrainment ( $\beta$ ) coefficients slightly. Several such changes are recommended in Wilson and Winkel (1982). For buoyancy rise, the alternate formulation is identical to Equation 2.17 except that  $\beta_1 = 0.93$  for high turbulence areas; the source for this change is listed as Briggs (1975), but it is not clear that that source directly suggests this change. The changes for momentum rise suggested by Wilson and Winkel (1982) are not clear. It appears that there may be typographic errors in the formulas as published, so that the exact changes being suggested are not clear. Furthermore, it appears that the source assumes constant  $\beta_j$  instead of  $\beta_j = \frac{1}{3} + \frac{u_s}{v_s}$ , which simplifies later calculations specific to the paper. Our best guess for the change suggested for momentum plumes is a change from  $\beta_j = 0.6$  (i. e., the same as  $\beta_1$  for buoyant plumes) to  $\beta_j = 0.66$ . Since there appear to be inconsistencies between the modifications in Wilson and Winkel (1982) and their cited source, Briggs (1975), we chose not to implement Wilson's suggestions.

Halitsky (1966) suggests the use of any appropriate rise estimate that is available; as with factory plumes, the calculation of rise is independent from the calculation of dispersion (other than that a plume height from some method is needed to calculate dispersion). For buoyant plumes Halitsky (1989) suggests a formula from Briggs (1989). He also suggests a relationship by Shandorov (also cited in Abramovich (1963)) that is useful for momentum-only jets (i. e., jets that are at the same temperature as the ambient air) that allows for variation in discharge angle. Although convenient, this relationship is not useful to us since we are interested in discharges which heat their surround and are thus at elevated temperature.

In the absence of a theory tailored specifically to buoyant transverse vents, we use the

same rise calculation methods that are used in CALPUFF for factory stacks. As these relations include momentum effects and stack size factors, it is not obviously unreasonable to apply them to roof vents.

#### 2.5.4 Transverse Vent Dispersion Calculations

Alternate methods for determining dispersion near buildings have also been suggested. Using the statistical theory of turbulence, a power law relation can be found for the local vertical dispersion coefficient (Wilson and Winkel, 1982).

An effort to fully study velocity and temperature profiles was begun in the late 1960's, but was cancelled before a model could be developed (Halitsky, 1970). According to the principal investigator of that effort, no further work in the area has been undertaken (Halitsky, 2000). However, Halitsky (1966, 1989) has proposed a simple model. He suggests matching a classic Pasquill plume in the far field to an empirically-defined jet region near the stack. In the jet region the plume has significant velocity, which decays to the freestream in the far field. This velocity decay is very rapid; it occurs over a distance of a few stack diameters. As the plume is rising rapidly in this zone, the jet region does not impinge upon the roof in ordinary circumstances; thus, in using this method, we are concerned only with the plume behavior in the fully-entrained region.

Halitsky empirically suggests that the fully-developed region develops at a downwind distance of

$$x_2 = (-10.66 + 11.07m^{0.673}) r_0 \quad (2.42)$$

where  $m = w_0/u_0$ ,  $w_0$  is the stack emission velocity,  $u_0$  is the freestream wind velocity near the stack exit, and  $r_0$  is the radius of the plume at the stack exit, which is set equal to the stack radius and has a constant concentration.

The assumed concentration distribution is shown in Figure 2.21. From the stack exit to some small distance downstream the concentration distribution is assumed to be trapezoidal changing to triangular. The characteristics of this region are not of particular interest for

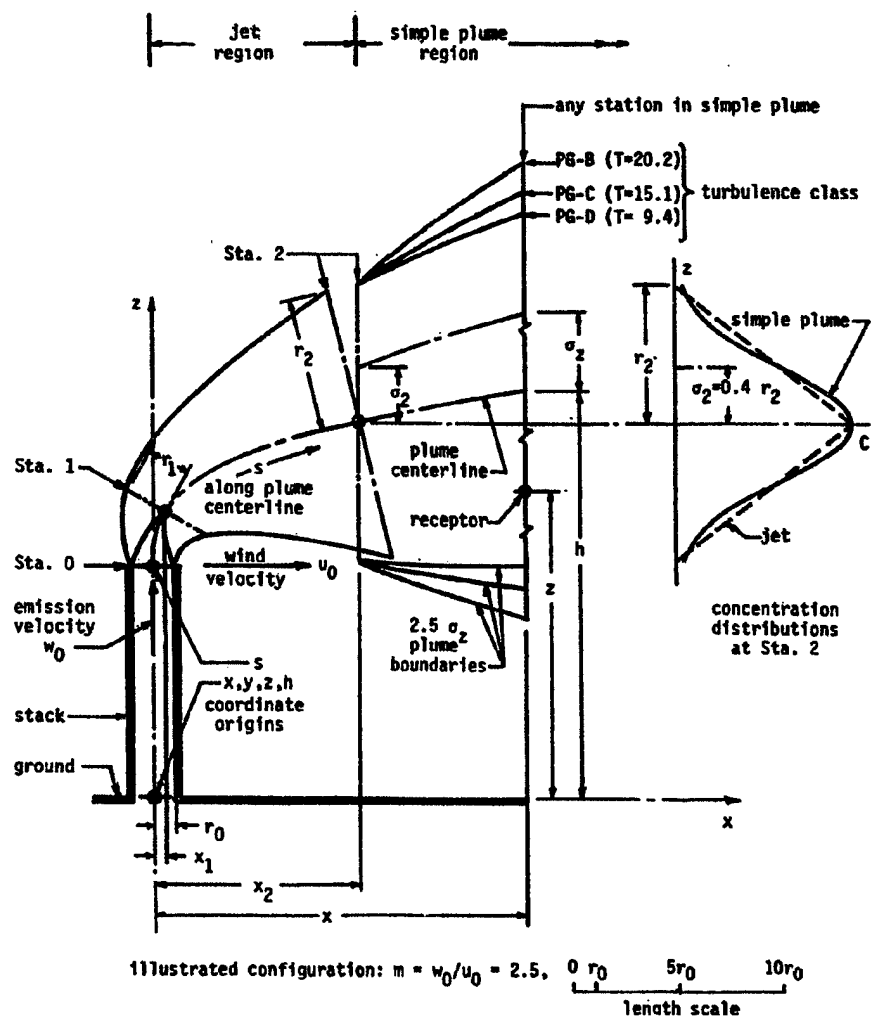


Figure 2.21: Assumed Plume Behavior for Transverse Jets (from Halitsky (1989))

SIG applications as the jet region is only on the order of a few stack diameters in length and the effluent is moving away from the roof or ground in the region.

The width of the triangle grows until station  $x_2$ , where it is matched to the simple Gaussian far-field plume. By this point, the plume has somewhat leveled out, and subsequent dispersion may bring it in contact with the roof or ground. At this station, the jet region has a triangular concentration distribution with a radius given by

$$r_2 = (-2.03 + 4.44m^{0.727}) r_0 \quad (2.43)$$

The jet region is matched to the Gaussian plume by setting  $\sigma_2 = 0.4r_2$ . This matching criterion, we note, alleviates the blooming problems associated with classic Pasquill plumes.

Figure 2.22 shows Halitsky's suggested parameterization of Turner's fits to Pasquill's stability curves. The horizontal dispersion curves are generated by the single relation

$$\sigma_y = 0.66e^{0.08(T_\sigma-30)}(x - x_2)^{0.92} + \sigma_2 \quad (2.44)$$

where  $x$  is the downwind distance in feet,  $T_\sigma$  is a stability parameter chosen as shown in Figure 2.22, and the resulting dispersion is in feet. (Halitsky (1989) uses  $T$  (as in Figure 2.22) for his stability parameter, which we modify to  $T_\sigma$  to avoid confusion with temperature.)

The vertical dispersion curves are similarly generated by

$$\sigma_z = 0.82e^{0.08(T_\sigma-30)}(x - x_2)^{0.80} + \sigma_2 \quad (2.45)$$

The horizontal dispersion is represented quite well by the single relation. The vertical dispersion is not matched as well for unstable conditions; the parameter  $T_\sigma$  is chosen so that the curves fit fairly well at a downwind distance of around 630 ft. Although Turner's fits match Pasquill's data better, Halitsky's approach has the benefit of being a function of a continuous stability parameter. This is helpful as the conditions in urban environments may not fit neatly in to the traditional six-level stability scheme; as appropriate, the stability parameter  $H$  can be varied to produce conditions intermediate between two stability categories.

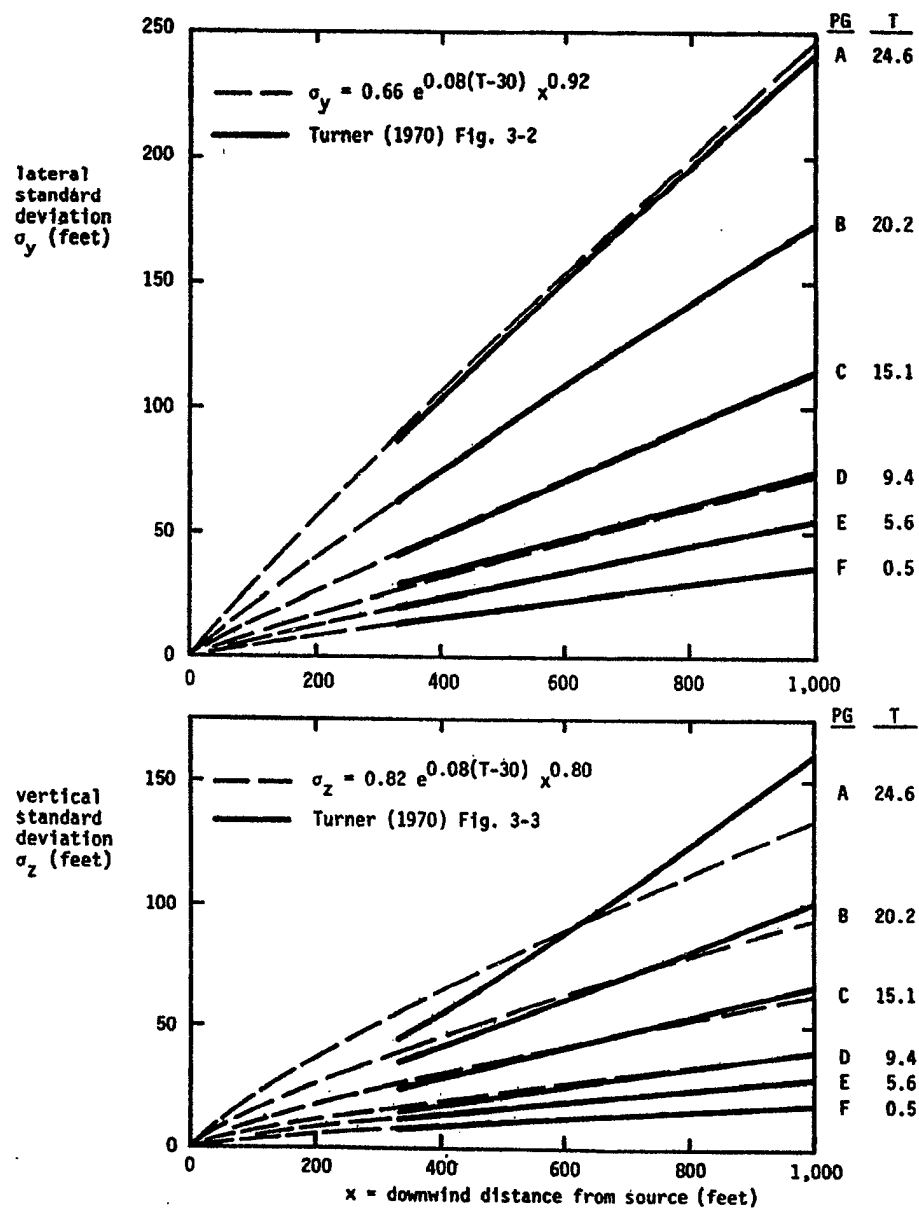


Figure 2.22: Dispersion Curves for Transverse Jets (from Halitsky (1989))

Plume rise for this type of jet plume can be calculated using any rise equation that includes a momentum or velocity factor. Thus, our existing rise equations (Equations 2.17, 2.18 and 2.20) are applicable.

Once the plume rise and dispersion are calculated, the concentration at a point  $(x, y, z)$  can be calculated from Equation 2.3. Alternatively, the dilution can be directly calculated as

$$D = \frac{2\sigma_y\sigma_z}{r_0^2 m} \exp\left(\frac{y^2}{2\sigma_y^2}\right) \left[ \exp\left(-\frac{(z-h)^2}{2\sigma_z^2}\right) + \exp\left(-\frac{(z-h)^2}{2\sigma_z^2}\right) \right]^{-1} \quad (2.46)$$

Practically, simple transverse jets are not significant for SIG applications. As was seen in Subsection 2.1.5, buoyancy rise dominates momentum rise even for small differences in temperature between the ambient air and the effluent. The rise is generally sufficient to ensure that the effluent only comes into contact with the ground or roof very far downstream, by which time it has cooled. Conversely, in momentum-dominated jets the effluent impinges on the ground more quickly, but there is no heating effect. Only in the case of impingement on nearby vertical surfaces do transverse jets cause significant convective heating, but Halitsky's model is not valid in these circumstances.

Nevertheless, because of its simplicity, we have described and implemented Halitsky's transverse jet model as an example of a simple user-written plume model. With some work, a more involved code could be developed to generate a puff-based plume using Halitsky's dispersion relations, which might then be more accurate than CALPUFF for modeling the direct observation of transverse jet plumes.

Likewise, the radiative heating effect of transverse jets when near roofs is very small. As the effluent often consists mostly of air and thus has low emissivity in most wavelengths with few clear spectral features, very little energy is radiated from the plume. That energy which is radiated tends to be reabsorbed quickly.

Radiative emission by stacks themselves, on the other hand, is significant. The stack will have a much larger emissivity than the emitted gas and thus will radiate far more energy. As with the ground warmed by a stack or any other warm object, this may be sensed,

typically in the long-wave infrared. As Halitsky's model does not deal with calculation of the temperature of the stack, the temperature must be specified separately in DIRSIG.

### 2.5.5 Axial Jets

Flows from axial jets have been examined by Abramovich (1963). For jets constrained by a wall, theory has been developed only for iso-thermal jets; although we err to some degree by doing so, we neglect the effect of temperature on such flows. In reality, the fact that the flow is heated will tend to cause it to disperse upward more rapidly than an iso-thermal jet due to buoyancy rise. Furthermore, the theory is developed for rectangular jets, which have the same height across a wide range of crosswind stations; we ignore this and use the theory for both round and rectangular vents. We have not quantified the error introduced by using the theory for conditions other than those for which it was developed. Such a study could be made by comparing the theoretical results to CFD results for a range of geometries. Despite these limitations, the theory produces results that are not obviously un-physical, and we thus retain the technique as an example of a simple user-written plume model that has some usefulness in SIG applications.

Because the flows are fast-moving (so that they have high Reynolds numbers), jet flows can be analyzed using boundary layer theory. Thus, a great deal of work originally conducted for aeronautical research is applicable here. Boundary layer theory breaks a flow down into two regions. In the outer layer, viscosity is assumed to be negligible, and classical fluid mechanics can be used to analyze the flow. Near a body, viscous forces and momentum forces are of the same order; this area is the boundary layer. A classic reference on fluid flow and boundary layer theory is Schlichting (1979).

### 2.5.6 Axial Jet Geometry

Figure 2.23 shows the geometry of a turbulent jet (emitted at point  $O'$ ) interacting with a plate (coincident with the  $x$  axis). The jet velocity,  $u_0$ , is greater than the freestream

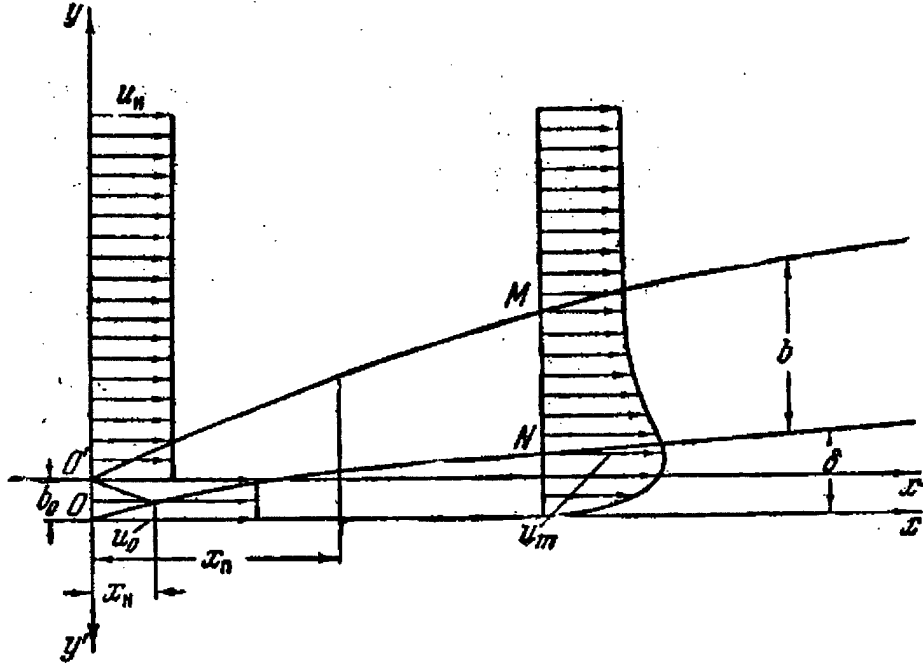


Figure 2.23: Geometry of Turbulent Jet Constrained by Plate (from Abramovich (1963))

velocity,  $u_H$ . In the first region, from  $x = 0$  to  $x = x_H$ , the jet boundary has not yet reached and interacted with the boundary layer of the plate. In this region, it is assumed outside the boundary layer that air is mixing into the jet in the same way it would if the wall was not there. It is assumed that the velocity outside the boundary layer but within the jet is  $u_0$  (which is somewhat counterintuitive;  $u_H$  would seem more appropriate). Within the boundary layer, a typical velocity profile for a turbulent boundary layer is assumed. This region is not really important for our purposes since the jet has no interaction with the wall.

The region from  $x = x_H$  to  $x = x_{II}$  is known as the transition region. By the end of this region, the potential core of the jet, the region in the center where the pressure is constant, has disappeared. The transition region is difficult to characterize, and is often

assumed to be of zero length.

The so-called main region of the jet begins at  $x = x_{\Pi}$ . (This point is shown in the figure; the subscript looks a bit odd because it is actually a Cyrillic “P” rather than a  $\Pi$ .) At this point, the velocity just outside the boundary layer is still  $u_0$ , as it was in the initial region. Beyond this point, this velocity,  $u_m$ , begins decreasing until it finally reaches  $u_H$  far downstream.

### 2.5.7 Theory of Axial Turbulent Jets

The theory for this case (Abramovich, 1963) assumes a velocity profile in the boundary layer of

$$\frac{u}{u_m} = \left(\frac{y}{\delta}\right)^{1/7} \quad (2.47)$$

where  $u$  is the velocity at a given height above the plate,  $u_m$  is the velocity just outside the boundary layer,  $y$  is the height above the plate, and  $\delta$  is the thickness of the boundary layer. A plot of this profile is shown in Figure 2.24. Close to the surface, the velocity increases very rapidly to a large fraction of the freestream value. This profile is often used in aerodynamic calculations because it is a reasonable representation of the profile in a turbulent boundary layer except in a very thin skin adjacent to the body (the viscous sublayer). The profile does not accurately model the viscous sublayer; the quantity  $\partial u / \partial y$  goes to infinity at  $y = 0$ . As the shear stress on the plate is proportional to this quantity, this velocity profile predicts infinite shear stress on the wall, which is clearly not realistic. Outside of the boundary layer, the theory assumes a velocity distribution identical to that in an unconstrained jet. By assuming that the upper edge of the jet expands as it would in the absence of the wall, and conserving momentum in the flow, the velocity at the edge of the boundary layer ( $u_m$ ) can be found. The procedure is outlined in Appendix D.

A sample result using this algorithm is shown in Figure 2.25; this figure shows one of the series found in Figure 11.6 in Abramovich (1963), except that an approximation made in Abramovich is not made here. The ordinate shows downwind distance, while the abscissa

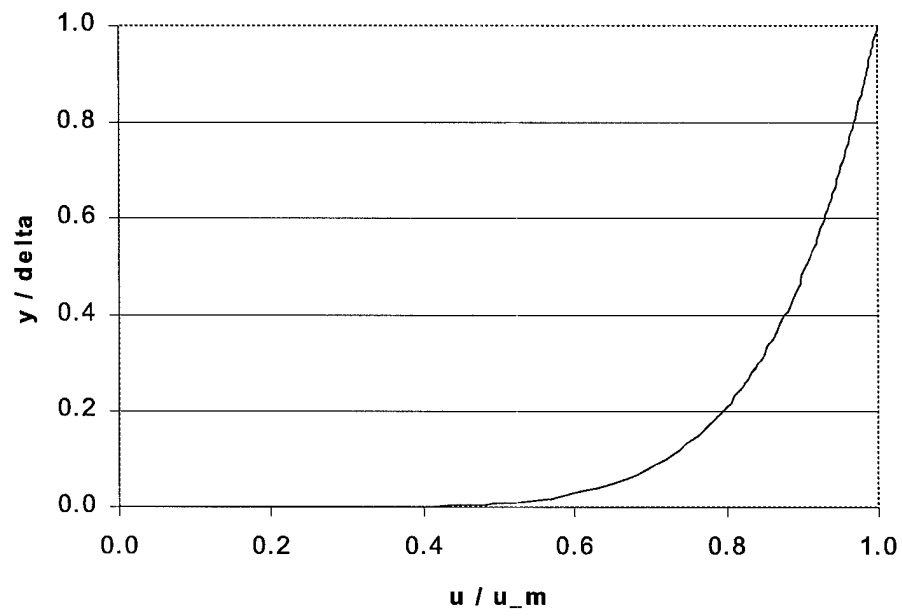


Figure 2.24: Velocity Profile in Turbulent Boundary Layer

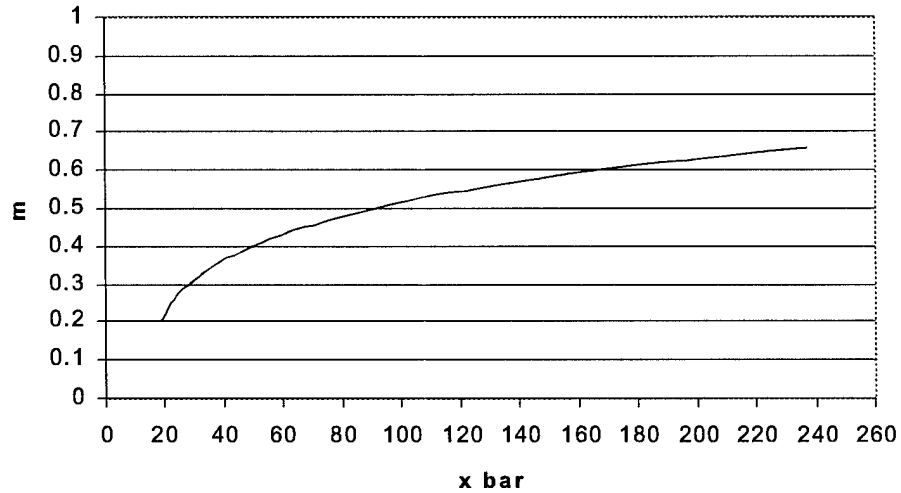


Figure 2.25: Velocity Ratio vs. Downwind Distance for Axial Jet

gives the ratio of the freestream velocity to the jet velocity (i. e., the ratio is less than one since the freestream is slower than the jet, and it increases as the jet becomes entrained in the freestream).

As seen in Figure 2.23, the theory assumes that the jet is emitted at the beginning of the plate. In reality, the boundary layer will already have a finite thickness by the time it reaches the vent and so will be thicker than predicted by this formula. However, since the boundary layer is actually relatively thin, the method still provides a reasonable estimate of the velocity at the edge of the layer however thick it might actually be. Since this estimate is the only product of the method we are using, the issue is not terribly significant.

Figure 2.26 compares the results of Halitsky's theory to Abramovich's theory. In the figure,  $\bar{x} = x/b_0$  where  $b_0$  is the initial vent height for the axial vent and the initial stack height for the transverse jet; thus, the two series are not strictly the same measure, but are comparable nondimensional measures.  $m$  is the jet velocity ratio, defined as the velocity

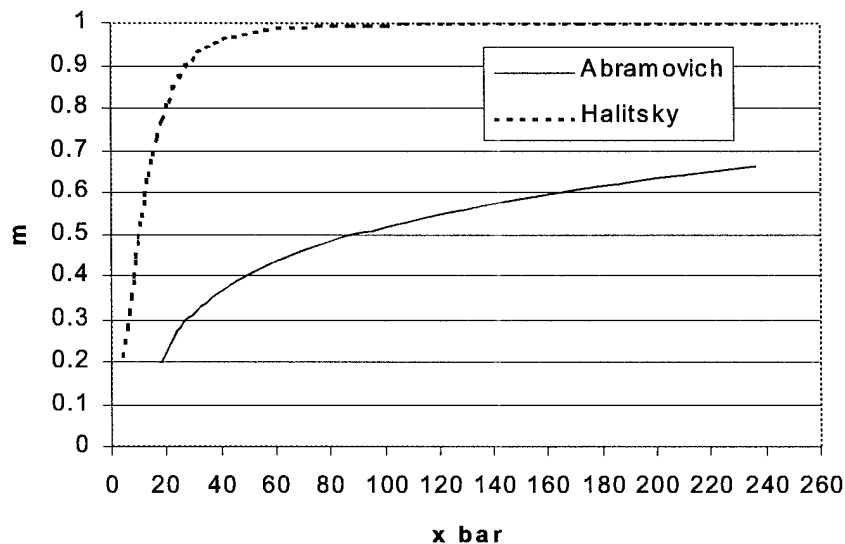


Figure 2.26: Velocity Ratio vs. Downwind Distance for Axial (Abramovich) and Transverse (Halitsky) Jets

of the jet freestream divided by the velocity of the jet. The most obvious feature of this plot is that Halitsky's theory postulates that the jet velocity decreases very rapidly to the freestream velocity. This phenomenon is consistent with the observation that plumes reach their final height within 10–30 stack diameters (Wilson, 1977); when the plume levels out, its initial momentum excess must have been dissipated.

### 2.5.8 Velocity and Temperature for Axial Jets

As Abramovich's theory is derived in two dimensions and for iso-thermal jets, there is no lateral velocity decay effect or temperature calculation intrinsic to the theory. However, basic jet theory provides methods for approximating velocity. Just as concentration and temperature follow similar profiles in factory plumes, there is some similarity in velocity and temperature profiles in jets, as shown in Figure 2.27 (Halitsky, 1966). Thus, it is logical

that there should be some similarity in form between concentration and velocity. Therefore, without validation, we assume that horizontal dispersion can be adequately characterized by Equation 2.44 for some value of the parameter  $T_\sigma$  (which may or may not be close to one of Halitsky's suggested  $T_\sigma$  values for any stability class), and with  $x_2 = 0$ . We use the calculated dispersion to attenuate the centerline velocity as calculated from Abramovich theory:

$$u(y) = u_m \exp^{-\frac{y^2}{2\sigma_y^2}} \quad (2.48)$$

where  $u_m$  is the centerline velocity from theory and  $y$  is the distance from the jet centerline to the point of interest in the direction perpendicular to the centerline.

Once the velocity is found in this manner, we assume that the velocity and the temperature dilute in the same manner; thus,

$$\frac{T_0 - T_H}{T - T_H} = \frac{u_0 - u_H}{u - u_H} \quad (2.49)$$

where the 0 subscripts denote conditions at the vent exit and the  $H$  subscripts denote ambient conditions. From this equation,  $T$  at a given point can be found.

This technique produces a teardrop-shaped warm spot on the ground, which is qualitatively correct. An example is shown in Figure 2.28. In the initial region the centerline temperature is unchanged; thus, the hot area grows as  $\sigma_y$  increases. Once the main region is reached the centerline temperature begins decreasing. This causes the hottest portion of the core to begin shrinking, although the portion that is warmed above ambient continues to expand, becoming more dilute with distance. (A linear feature of different material cuts across the scene and thus has a slightly different surface temperature.)

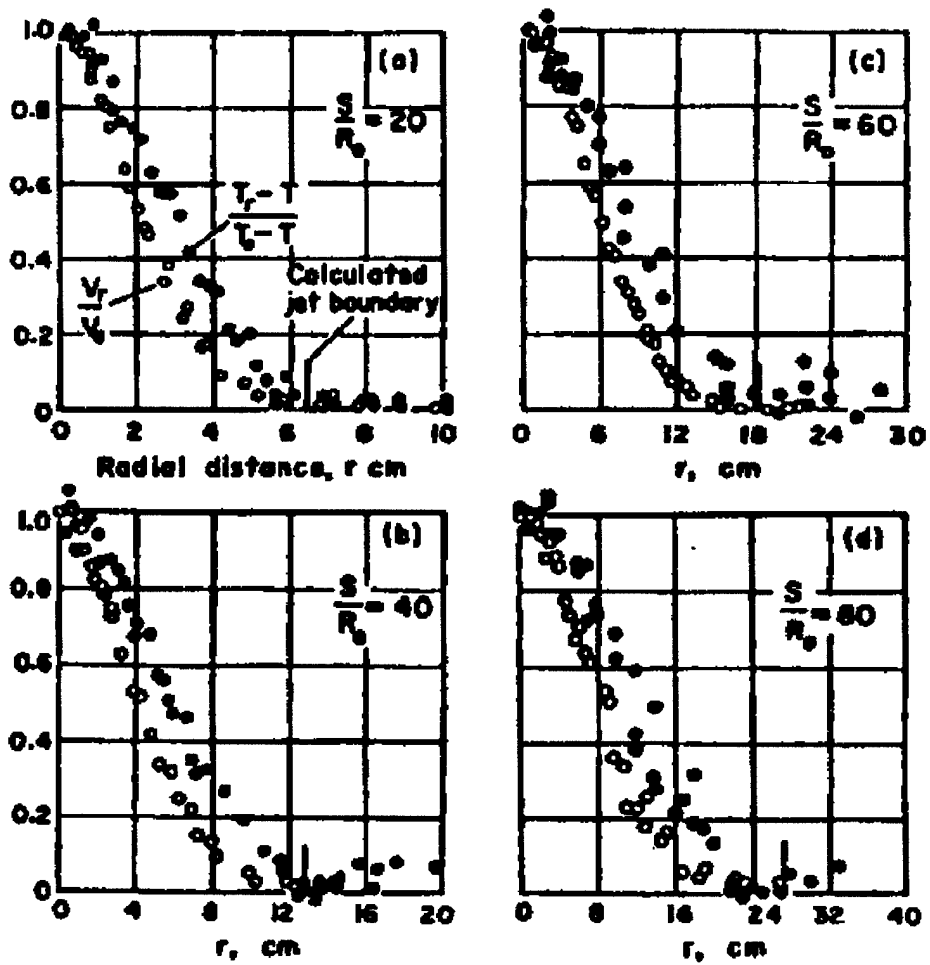


Figure 2.27: Nondimensional Radial Temperature and Velocity Profiles (from Halitsky (1966))

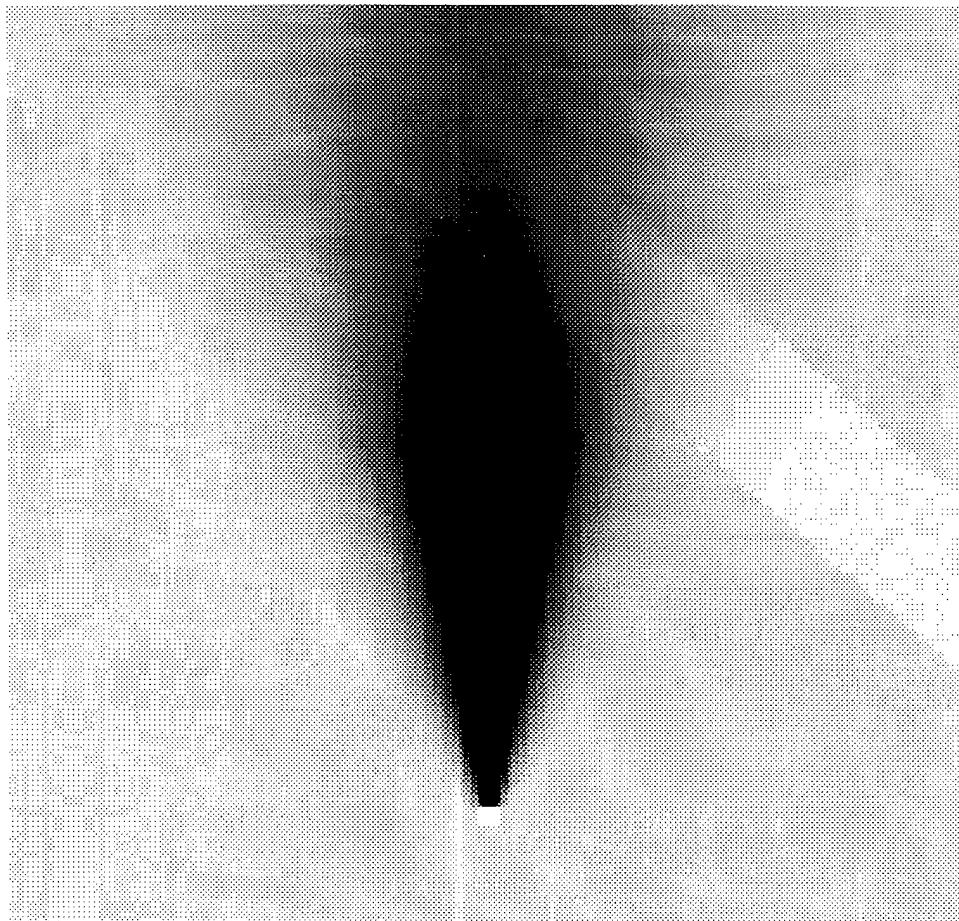


Figure 2.28: Temperature Map for Axial Vent

### 2.5.9 Downward Vents—Foundations of Turbulent Fluid Motion

As there is no satisfactory theory for downward-discharging vents and other complex fluid motion, we must obtain solutions for these cases from Computation Fluid Dynamics (CFD). An in-depth treatment of the theory of turbulent flow is far beyond the scope of this work; an excellent book on the subject is Wilcox (1994). Any user who wishes to run new CFD geometries is urged to consult this book and others. The brief description that follows is intended for those who are content to use the CFD solutions we have run as-is.

The fundamental equations describing fluid flow are the continuity and Navier-Stokes equations. For incompressible, turbulent flow, the continuity equation, which ensures conservation of mass, is

$$\frac{\partial U_i}{\partial x_j} = 0 \quad (2.50)$$

The Navier-Stokes equations, which, for turbulent flow, characterize the average fluid flow, are

$$\rho \frac{\partial U_i}{\partial t} + \rho U_j \frac{\partial U_i}{\partial x_j} = -\frac{\partial P}{\partial x_j} + \frac{\partial}{\partial x_j} (2\mu S_{ij} + \tau_{ij}) \quad (2.51)$$

Note that this equation gives properties that are averaged over some time; turbulence is a chaotic phenomenon, and calculation of exact properties is not possible and would not be particularly relevant, as miniscule changes in initial and boundary conditions cause the specific details of a flow to change dramatically.

For those not familiar with the notation, Equations 2.50 and 2.51 are written in Einsteinian notation; in this scheme, often used by fluid dynamicists, any index that is repeated within a term (as  $j$  is in the second term of Equation 2.51) implies summation over that index. Any index that is not repeated is a free index, which is not summed, but which takes different values in different instances of the equations. Thus, for two dimensional flow, there are two Navier-Stokes equations; one is for the  $x$ -velocity  $U_1 = u$ , and the other for the  $y$ -velocity  $U_2 = v$ . The other variables in the equations are the spatial variables  $x_1 = x$  and  $x_2 = y$ , the pressure  $P$ , the viscosity  $\mu$ , the stress tensor  $S_{ij}$ , and the Reynolds tensor  $\tau_{ij}$ .

Except for the addition of  $\tau_{ij}$ , the Reynolds-averaged turbulent Navier-Stokes equations are identical to the laminar equations. The Reynolds tensor characterizes the eddy viscosity of the flow. Just as molecules resist sliding across one another and produce a stress opposing the relative motion, larger eddies ranging in size from a few molecules to macroscopic pockets also resist relative motion. This effectively increases the viscosity of the fluid.

An entire branch of the study of fluid flow is devoted to modeling the Reynolds tensor. Within the past few years, it has become conceivable to simulate the flow on a molecular level. Such methods are not, however, well-developed except for very basic flow configurations. Thus, the empirical methods developed since the 1930s are used for more complex flows. Two models well-suited to downward vent flow are the  $k - \epsilon$  and  $k - \omega$  models. These two models are the most popular turbulence models in wide use. The details of the methods, and several others, are given in Wilcox (1994), an excellent reference on computational modeling of turbulence.

The Reynolds tensor is related to the kinetic energy contained in the turbulence ( $k$ ) and the dissipation of turbulence ( $\epsilon$  or  $\omega$ ) through a system of coupled partial differential equations. The dissipation is expressed either directly ( $\epsilon$ ) or as dissipation per time ( $\omega$ ). The two are related, as  $\epsilon \sim \omega k$  and knowledge of either along with knowledge of  $k$  is equivalent to knowledge of the average length scale of turbulent fluctuations in a neighborhood. Which form is found first is a function of which model is used. In general, the  $k - \epsilon$  model is less sensitive to inaccuracies in the estimates of  $k$  and  $\epsilon$  at the flow boundaries, but does not perform well in cases where the flow separates from surfaces. In these cases, which would be found, for example, if the wind flow around buildings or over buildings were calculate, the  $k - \omega$  model is preferred. Forms of both models are available in the WIND CFD code that we have chosen.

It is pointed out very strongly in Wilcox (1994) that, although the equations behave somewhat like the actual production and dissipation of turbulence at the molecular level do, the constants are strictly empirical, and the equations are only conceptually representative

of the actual processes. As with Briggs' macro-scale plume rise calculations, the extra equations close the problem so that it is no longer underconstrained. It is often necessary to tweak the constants for improved correlation with experiment, and coefficients are often optimized for a particular flow problem. (It is largely for this reason that we have not considered using CFD techniques for factory stacks—doing so would require the use of constants tailored to the application, which may not be available for all scales of stacks. Besides requiring enormous run times, these constants would likely not produce results significantly better than those obtained from using traditional methods.) General-purpose values of the coefficients are the default values used in the WIND code. As WIND is oriented toward aerodynamic simulation, the constants are most reasonable when there is a dominant freestream airflow with perturbations caused by objects in the flow as with downward pointing jets (provided the mass of injected air is small compared to the mass of the freestream air) and to wind flow around buildings (which we have not presently considered but which is of some interest for SIG applications).

As will be discussed in more detail in Subsection 3.3.3 and Appendix F, we have implemented a method for extracting results from WIND and using them with DIRSIG's thermal model to predict temperatures within vent flows.

## Chapter 3

# Research Approach

The theory outlined in the previous chapter describes how factory stacks and roof vents can be modeled. In this chapter, the problem flow is broken down into basic units.

### 3.1 Generic Plume Interface (GPI) Outline

All of the models implemented in our research communicate with DIRSIG using a Generic Plume Interface (GPI). This interface, as shown by the four example clients that will be described in the following sections, enables diverse models to communicate with DIRSIG using a simple and generic protocol. The four GPIs to be discussed are shown in Figure 3.1. On the left is the CALPUFF model, used primarily for large stacks. On the right are the three “vent” models: the Halitsky model for transverse vents, the Abramovich model for axial vents, and the CFD model for downward or other arbitrary vents.

#### 3.1.1 Overall GPI Flow

DIRSIG communicates with a GPI at three steps of the image generation process: initialization, ray tracing, and exit. The overall sequence is shown in Figure 3.2.

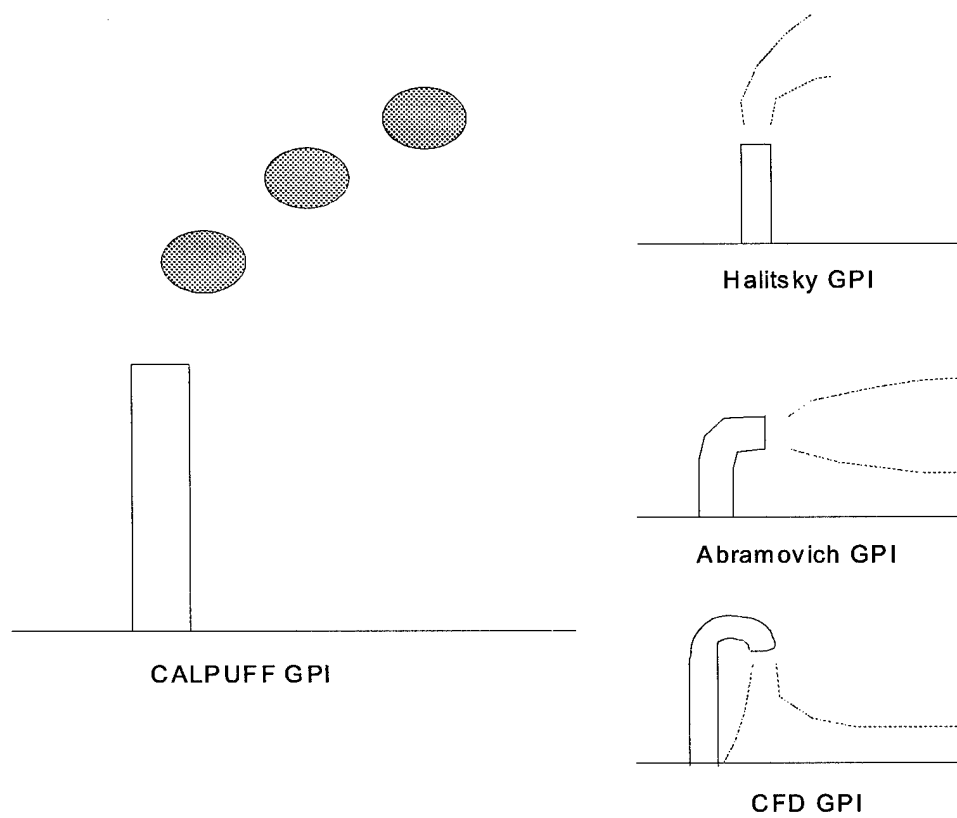


Figure 3.1: Schematic of Four Types of GPI Clients

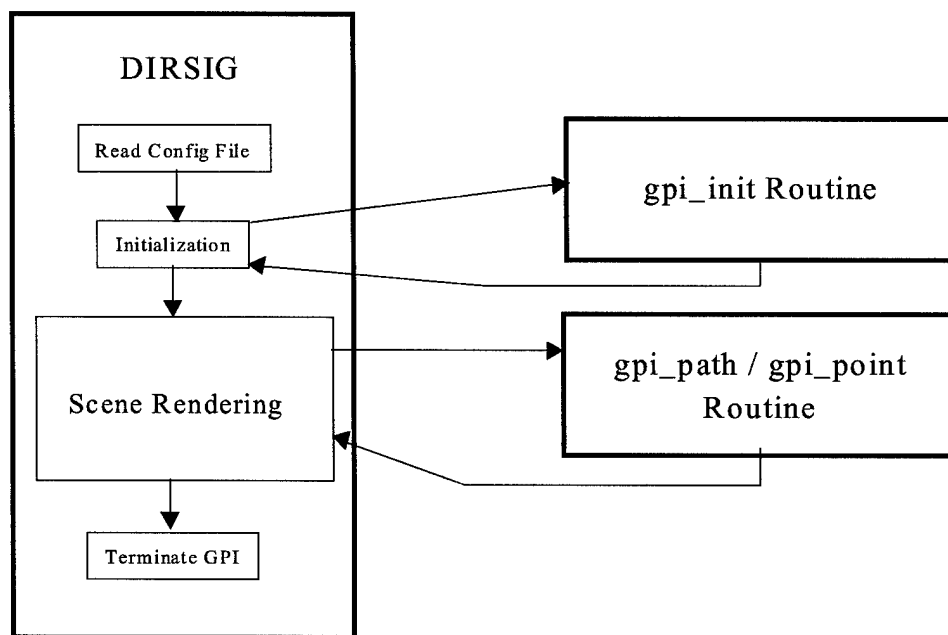


Figure 3.2: Overall Flow for Generic GPI Client

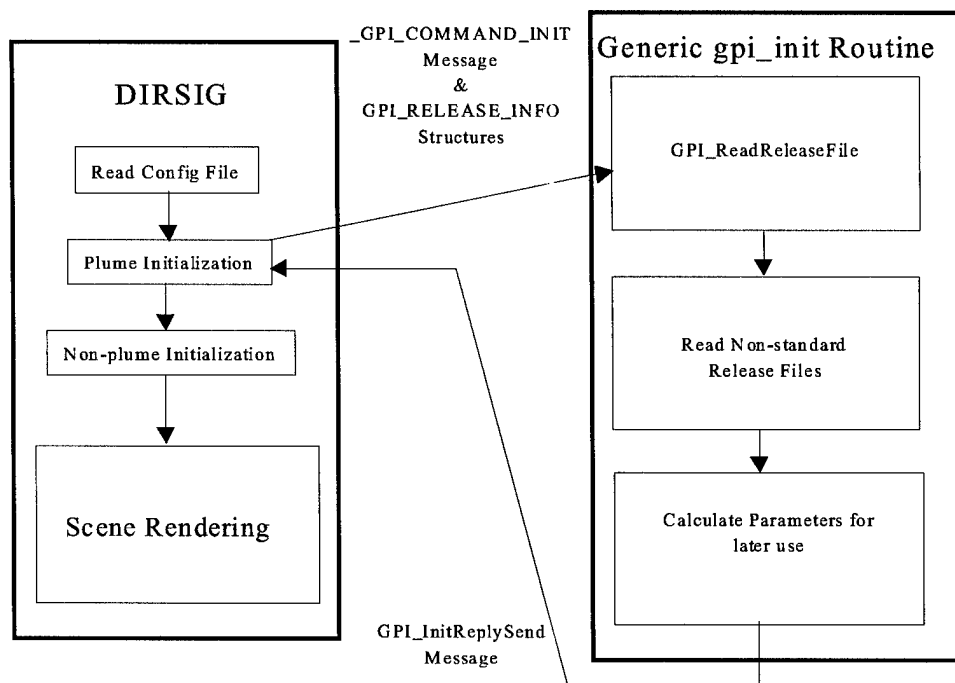


Figure 3.3: Problem Initialization Flow for Generic GPI Client

## Initialization

Before rendering, DIRSIG reads its configuration file and starts the appropriate GPI clients. Generally, the first task of the client is to read the DIRSIG release file, which contains stack or vent geometry, location, release parameters, etc. Although this is technically the job of the user-written client, a standardized function for reading the file is available, and the contents of the structure filled as the file is read are intended to be generic. Because it is standardized, we conceptually portray the release structure as being provided to the client by DIRSIG. After receiving the default release parameters, the initialization module performs any other tasks that may be necessary including reading of model-specific input files and calculation of parameters that do not change as a function of time or location. This flow is shown in Figure 3.3.

## Ray Tracing

After initialization of the plume model and other DIRSIG initialization tasks, DIRSIG begins rendering the image. The GPI modules accept information about each ray being traced and report the influence of the plume on the ray. Two types of interaction are currently implemented. Effects that are dependent on both the beginning and end of the ray are queried using a `gpi_path` request. Point effects are queried using a `gpi_point` request. Currently, path queries are used for factory stacks since the plumes absorb and emit continuously along paths of interest. Point queries are currently used for roof vents, for which the effect on a surface is independent of the path length or angle. Although our existing models calculate responses for only path or only point queries and return null responses for the other, both could be used meaningfully in the same model.

## Exit

GPI models continuously wait for commands in the background following initialization. Thus, the variables and structures used in rendering are maintained in memory, making ray tracing more efficient. Thus, when DIRSIG is finished rendering the image, the GPI clients must be explicitly shut down. Under normal circumstances DIRSIG automatically performs this task, but clients occasionally remain running if DIRSIG encounters problems and terminates abnormally.

## 3.2 Factory Plume Modeling—CALPUFF GPI Client

The first, and most important, area of this research is the modeling of factory plumes. The modeling process consists of two parts. In the first part, overall data about the scene is provided by DIRSIG to the GPI. This initialization results in an array of puff locations, indexed by time and species, which is then used in ray tracing. As DIRSIG renders the scene, the geometry of each ray is passed to the plume interface, which traces it through

the puff field and returns a series of concentrations and temperatures along the plume.

### 3.2.1 CALPUFF `gpi_init` Flow

For the CALPUFF GPI client, much of the plume modeling work is done at initialization. The routine receives from DIRSIG two structures containing problem parameters. The first, a `GPLRELEASE_INFO` structure, is intended to be generic across all GPI client types and is mostly filled by the GPI prior to initialization of the CALPUFF client. The second, a CALPUFF-specific structure of type `CALPUFF_INFO`, is filled by `gpi_init` and contains the data that will later be required as rays are traced.

At present, CALPUFF (probably preceded by CALMET) must be run prior to execution of DIRSIG. The client expects to find a `debug.dat` file in the directory from which DIRSIG is run.

The basic flow through the client is shown in Figure 3.4. The AR models used for wind modeling are discussed shortly in Subsection 3.2.3. The portions of the MIND algorithm discussed in Subsection 2.2.5 are used to smooth the wind and calculate the instantaneous diffusion. The time-varying smoothed wind is used to perturb the puff locations calculated by CALPUFF. The rise for each puff is calculated as a function of chemical species.

At completion of the `gpi_init` procedure, the `CALPUFF_INFO` structure is filled with the properties (location, mass, chemical species) of all puffs for the entire sequence of times of interest.

### 3.2.2 CALPUFF `gpi_path` Flow

As rays are traced, the GPI passes information on each ray to the GPI client, where it is processed by the `gpi_path` routine. As shown in Figure 3.5, the routine finds the point of closest approach along the ray to each puff (for the time at which the ray is traced). It then determines whether each puff is significant and determines the overall portion of the ray which interacts with significant puffs. It then steps along this segment of the ray,

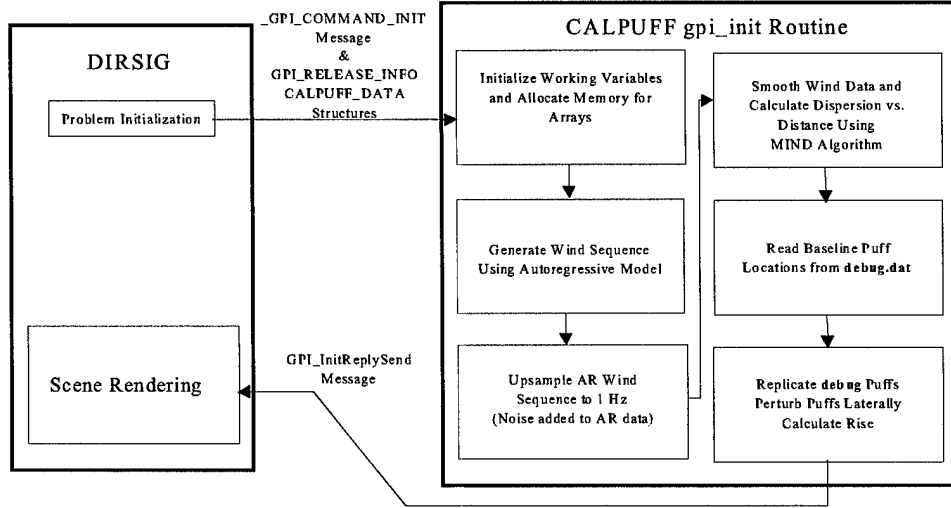


Figure 3.4: Problem Initialization Flow for CALPUFF GPI Client

calculating temperatures and concentrations at each step. The methods for these steps are discussed in this subsection. The results of the tracing are returned in a structure of type `GPI_PATH_REPLY`.

After receiving the reply, DIRSIG calculates the overall transmission and thermal emission for the path.

### Point of Closest Approach

A distance of closest approach is calculated between the ray and each puff. The line segment connecting the two ends of the ray,  $(x_1, y_1, z_1)$  and  $(x_2, y_2, z_2)$  is parameterized as

$$\begin{aligned}
 x &= x_1 + s(x_2 - x_1) \\
 y &= y_1 + s(y_2 - y_1) \\
 z &= z_1 + s(z_2 - z_1)
 \end{aligned} \tag{3.1}$$

where  $0 \leq s \leq 1$ .

The squared distance from the arbitrary point  $(x, y, z)$  to the center of a puff  $(x_p, y_p, z_p)$

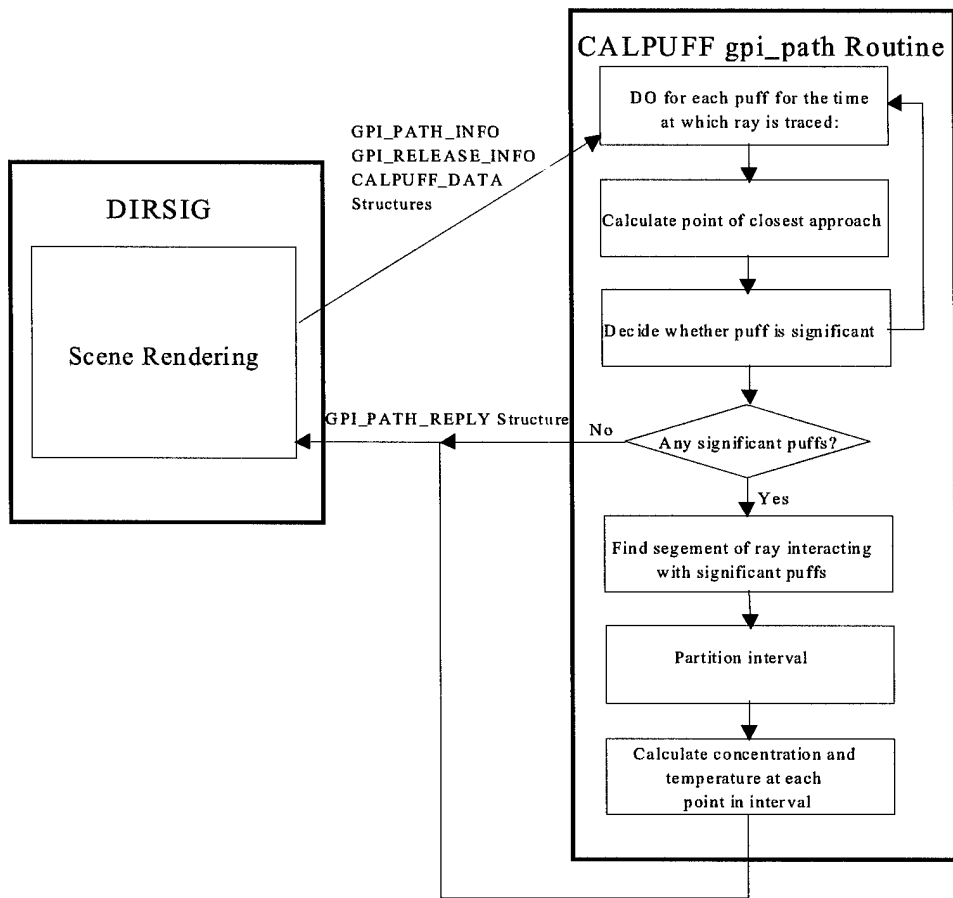


Figure 3.5: Ray Tracing Flow for CALPUFF GPI Client

is then

$$d^2 = (x_1 + s(x_2 - x_1) - x_p)^2 + (y_1 + s(y_2 - y_1) - y_p)^2 + (z_1 + s(z_2 - z_1) - z_p)^2 \quad (3.2)$$

Differentiating  $d^2$  and setting it equal to zero gives

$$\begin{aligned} \frac{d(d^2)}{ds} = 2 \{ & [(x_1 - x_p) + s(x_2 - x_1)](x_2 - x_1) + \\ & [(y_1 - y_p) + s(y_2 - y_1)](y_2 - y_1) + \\ & [(z_1 - z_p) + s(z_2 - z_1)](z_2 - z_1) \} = 0 \end{aligned} \quad (3.3)$$

so that the parameter at the point of closest approach is

$$s = - \frac{(x_1 - x_p)(x_2 - x_1) + (y_1 - y_p)(y_2 - y_1) + (z_1 - z_p)(z_2 - z_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (3.4)$$

from which the location of closest approach and distance at closest approach can be found via Equations 3.1 and 3.2.

To determine whether a puff is optically significant for a given ray, we compare the distance of closest approach to the size of the puff. If  $d \geq 4.0 \max(\sigma_y, \sigma_z)$ , we neglect the puff. An example is shown in Figure 3.6, where the puffs in dark gray are those within four standard deviations of the ray.

### Segmenting of Ray Path

Next, the region of the ray which does not interact with any puffs is excluded; this process is shown in Figure 3.7. The distances from the puffs to the ray are greatly exaggerated.

The values of  $s$  at closest approach to each significant puff are sorted. The smallest  $s$  represents the first puff encountered by the ray, and the largest  $s$  represents the last puff. A length  $4.0\sigma_{max}$ , where  $\sigma_{max}$  is the largest sigma for any puff in either the  $y$  or  $z$  directions, is subtracted from the smallest  $s$  value and added to the largest  $s$  value. This gives the segment of the ray that interacts significantly with any puff. These limits have been chosen based on simple test cases so that the reported concentrations have a few, but not too many, points with nearly zero concentration on either end. Also, it is pointed out that, as currently

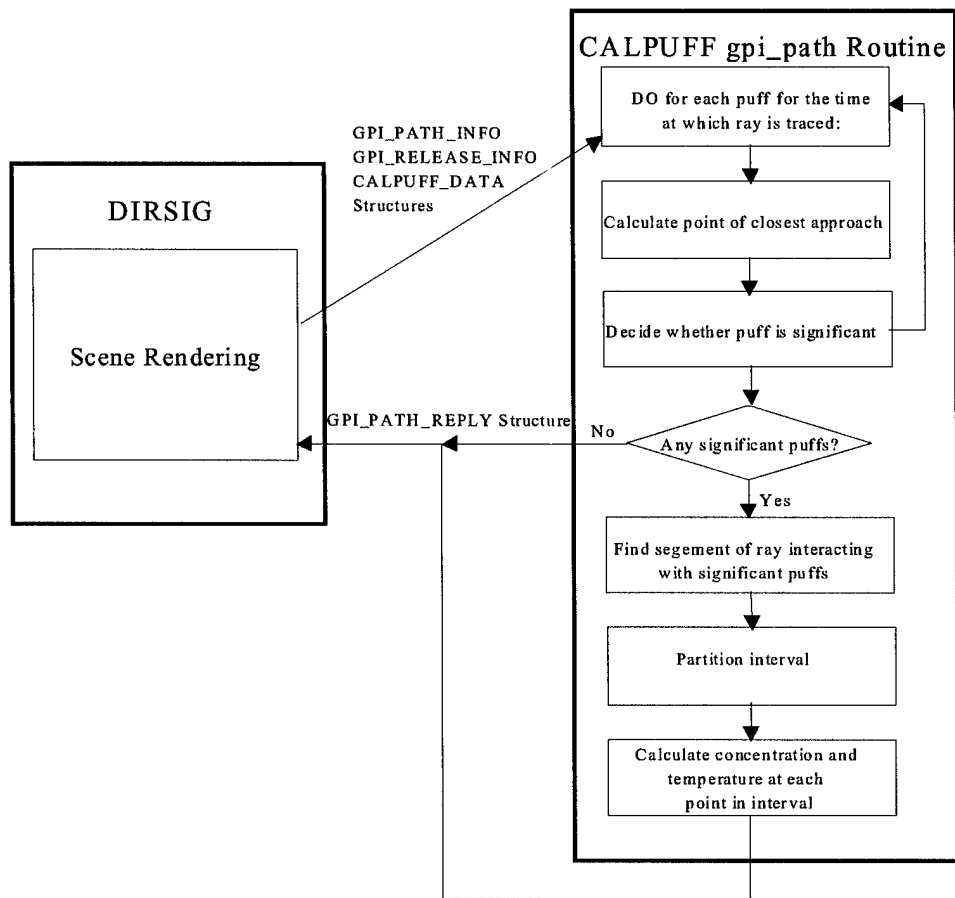


Figure 3.6: Identification of Significant Puffs for a Ray

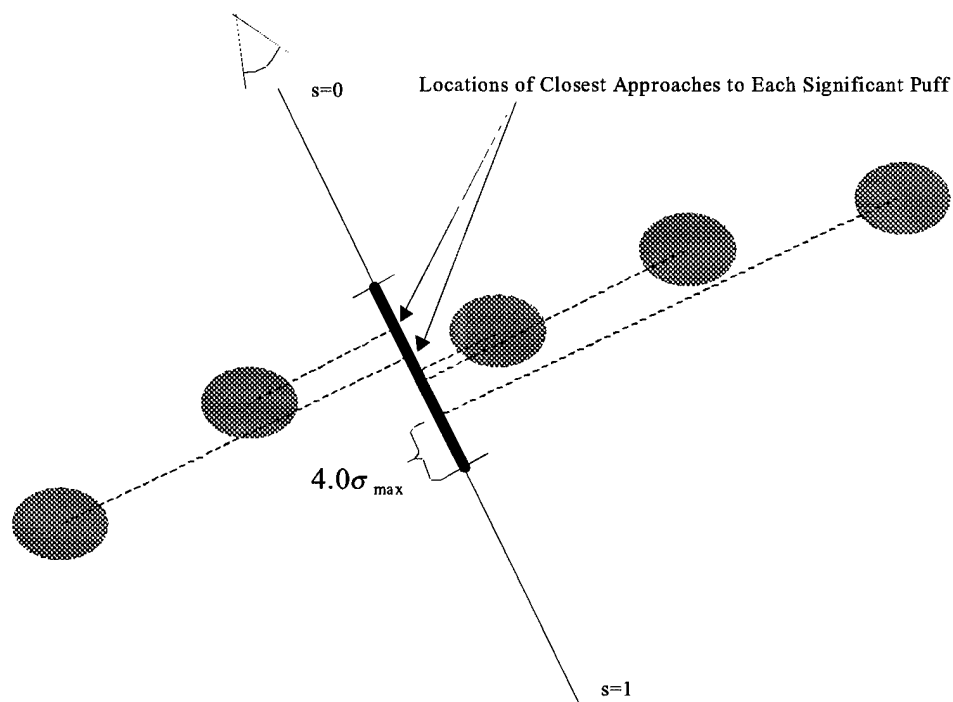


Figure 3.7: Segmentation of Ray

implemented, the region of "significant" concentration spans from the first entrance into the plume to the last exit. Therefore, if a ray interacts with a plume twice or with multiple plumes, large regions with zero concentration will be reported between the interactions.

### Calculation of Concentration and Temperature

This region is then segmented into a number of intervals. A variety of strategies are available for choosing the intervals. We have chosen to space the points evenly with a spacing of approximately 5 cm each, but with a maximum of 1024 points. Although we have chosen an even spacing, the GPI is capable of reporting uneven intervals and DIRSIG is capable of using them.

Given each of the evenly-spaced  $s$  values,  $(x, y, z)$  values are found using Equation 3.3. The concentration at each point is then given by

$$C = \sum_{i=0}^n C_i \quad (3.5)$$

where the  $C_i$  for each puff are given by

$$C_i = \frac{Q}{2\pi\sigma_{yi}^2} g \exp \left[ -\frac{(x - x_{pi})^2}{2\sigma_{xi}^2} \right] \exp \left[ -\frac{(y - y_{pi})^2}{2\sigma_{yi}^2} \right] \quad (3.6)$$

with

$$g = \frac{2}{(2\pi)^{1/2}\sigma_{zi}} \left\{ \exp \left[ -\frac{(z - z_{pi})^2}{2\sigma_{zi}^2} \right] + \exp \left[ -\frac{(z + z_{pi})^2}{2\sigma_{zi}^2} \right] + \right. \\ \exp \left[ -\frac{[2h - (z - z_{pi})]^2}{2\sigma_{zi}^2} \right] + \exp \left[ -\frac{[2h - (z + z_{pi})]^2}{2\sigma_{zi}^2} \right] + \\ \left. \exp \left[ -\frac{[2h + (z - z_{pi})]^2}{2\sigma_{zi}^2} \right] + \exp \left[ -\frac{[2h + (z + z_{pi})]^2}{2\sigma_{zi}^2} \right] \right\} \quad (3.7)$$

(Note that Equations 3.6 and 3.7 are functionally identical to Equations 2.26 and 2.28.)

Once the concentration at each point is known, the temperature at each point is easily calculated using Equation 2.14, since the concentration at stack exit, ambient temperature, and stack gas temperature are known.

### 3.2.3 Development of Wind Models

As discussed in Section 2.3, an AutoRegressive Integrating Moving Average (ARIMA) model is used to characterize the wind process. The procedure developed for this particular application is a combination of advice from Box and Jenkins (1976), Bowerman and O'Connell (1987), and Wood (2000).

Two statistics packages were used to develop the models. The first, Minitab, is a Windows-based general statistics program that specializes in interactive development of models. The second, R (R Development Core Team, 2001), is a data manipulation language (based on another data language known as S) for which many statistics libraries, including time series analysis, have been built. Although it includes a command console feature, its main strength is in its ability to do batch processing of data. The R system and the "tseries" library are Free Software (published under the GNU Public License) and are thus available to all.

#### **Dataset st712c**

The first example developed here is a dataset known as "st712c" in O'Neill (1996b) and "S712c" in Peterson et al. (1999). The dataset contains 50 minutes of data at 1 Hz (3000 points). The stability class for the data is estimated as class C (O'Neill, 1996b) (Peterson et al. (1999) lists it as class D, but, according to Environmental Protection Agency (2000), which outlines the method used to estimate the class, it should be class C). They were acquired on July 12, 1996 at Galen, Montana. First, a time series plot of the data is analyzed. The speed and direction are shown in Figures 3.8 and 3.9. The speeds are reported in meters per second and the directions in degrees with 360 deg added to all directions from north to east (0 to 90 deg).

The two plots show the recorded speed (in m/s) and direction (from which the wind blows, in deg) for the entire 50 min data set. For the direction, 360 deg has been added to all winds in the range 0 to 90 deg, to allow directions in the northwest quadrant to be

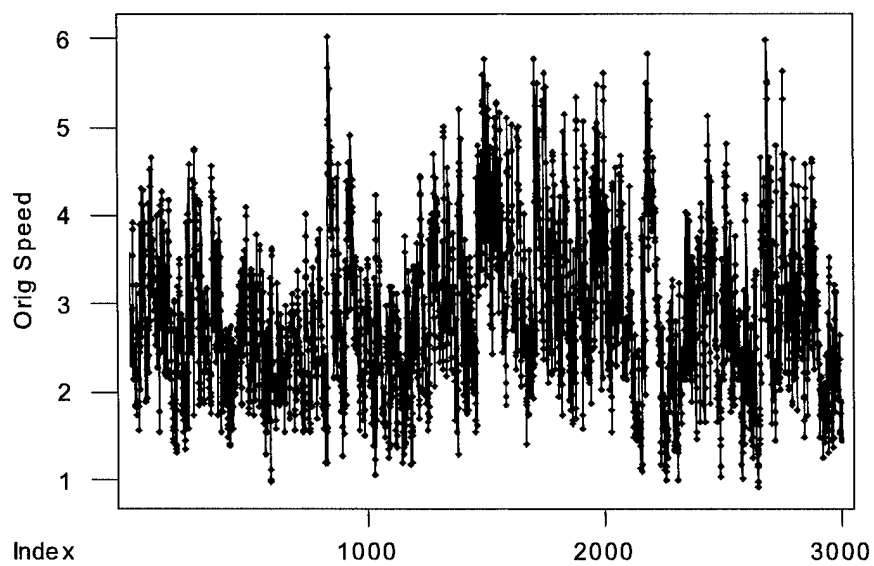


Figure 3.8: Speed History for Dataset st712c

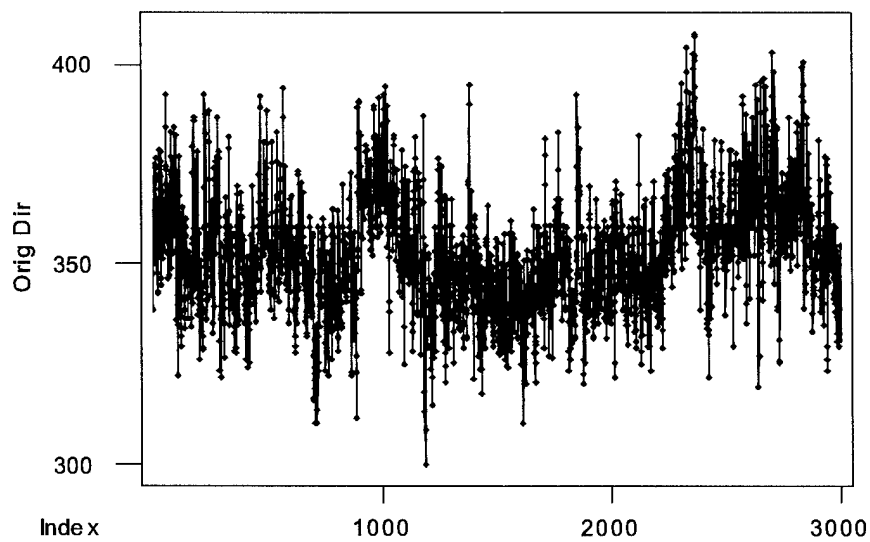


Figure 3.9: Direction History for Dataset st712c

easily subtracted from directions in the northeast quadrant. At first glance, there is no obvious deterministic trend to the data. There are some gradual fluctuations that might require differencing, but, without a physical reason for these fluctuations, they are best left unmodeled for the time being.

First, the ACF and PACF for the direction data are calculated. The direction is analyzed first because the MIND model primarily uses the wind direction in modeling meander. The ACF and PACF are shown in Figures 3.2.3 and 3.10.

The vertical bars show the ACF and PACF at each lag. The dashed lines running horizontally represent confidence limits for the lags; in this case, spikes that pass through the lines are statistically significant with 95% confidence. At the bottom of each figure are lists of the calculated values and T statistics at each lag.

Two significant observations are apparent from the graphs. First, the ACF appears to die down, while the PACF seems to cut-off very quickly. This hints that the wind direction process is predominantly autoregressive. Second, the ACF dies down extremely slowly; it is significant at all lags shown on the chart. Naively modeling the raw data would require a very high-order model with an increased computational burden; such a model would also tend to fit itself to any noise or random characteristics of this particular dataset and would not be a good model for slightly perturbed conditions. A solution to this problem, suggested by Dr. Hubert Wood of the Center for Quality and Applied Statistics at the Rochester Institute of Technology, is to average the data to reduce the sampling frequency. By doing this, we seek to find a sampling frequency of the same order as the frequency at which the quantity being measured actually changes. An alternative technique would be to retain one data point, drop several, retain one, etc., but this might tend to accentuate noise in the instrument.

Figures 3.2.3 and 3.12 show the reduced data sets after averaging for periods of 10 and 30 seconds, respectively. With 10 second averaging, the time series still has most of its character. With 30 second averaging, it seems at first glance that significant structure has

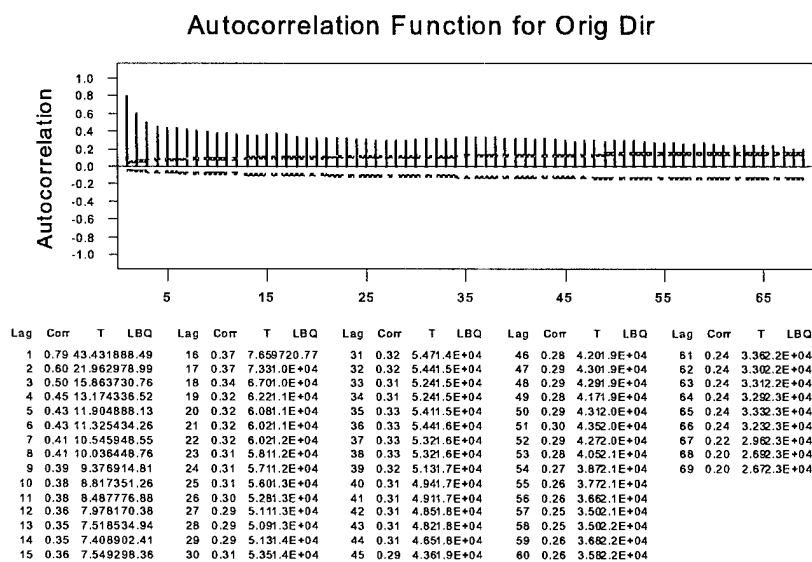


Figure 3.10: Autocorrelation Function (ACF) for Wind Direction in Dataset st712c

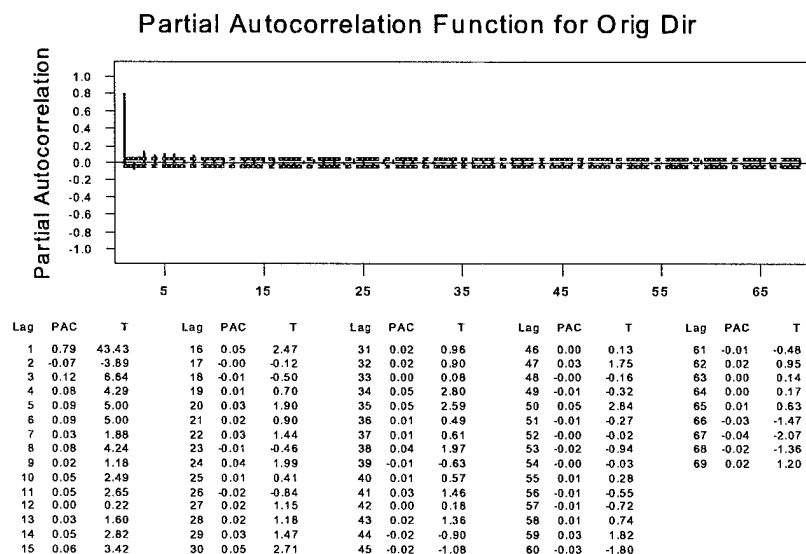


Figure 3.11: Partial Autocorrelation Function (PACF) for Wind Direction in Dataset st712c

been lost to the averaging.

The ACFs for the two cases are shown in Figures 3.14 and 3.15. With 10 second averaging, the ACF is significant out to 11 lags; with 30 second averaging, it is significant only out to 3 lags. Either qualifies as dying out, suggesting a pure autoregressive model.

The PACFs for the two cases are shown in Figures 3.16 and 3.17. In both cases the PACF is significant to about four lags (with a spike again at lag 12 in the case of 10 second averaging). This cutting-off behavior also indicates that the wind direction process is autoregressive and of approximately fourth order. It is interesting that both averaging periods yield the same order of process despite different behavior. For 10 second averaging, all four lags are significant; for 30 second averaging, lags 2 and 3 are not significant. Thus, increasing the averaging period from 10 to 30 seconds affects the PACF, but not the first-order conclusion derived from it.

As the 10 second-averaged data has a manageable degree of correlation and preserves more detail of the original data than the 30 second averaging, an autoregressive model is developed for the 10 second averaging. Although the PACF suggests that a fourth-order model is appropriate, we begin with a first-order model and progressively add terms, ensuring that each additional term is statistically significant.

First, a first-order model with no constant term was tested. Minitab converged to a model and reported parameters, but this model is deficient. When evaluating an ARIMA model, Minitab uses it to “back forecast” the original data, or estimate earlier values based on the last ones in the data set. In this case, the residual error between the back forecasts and the actual data did not die out. In general, this indicates that there is a non-stationarity in the data that has not been accounted for. Minitab also reported this error for all orders of autoregressive models through order five that did not include a constant term.

Adding a constant term gives an acceptable AR(1) model for which the back forecasts die out. The model is  $z_t = 118.427 + 0.6646z_{t-1} + a_t$  where  $a_t$  is a normal random variable. (Note that the form of the model given by Minitab gives  $z_t$  directly, including the mean,

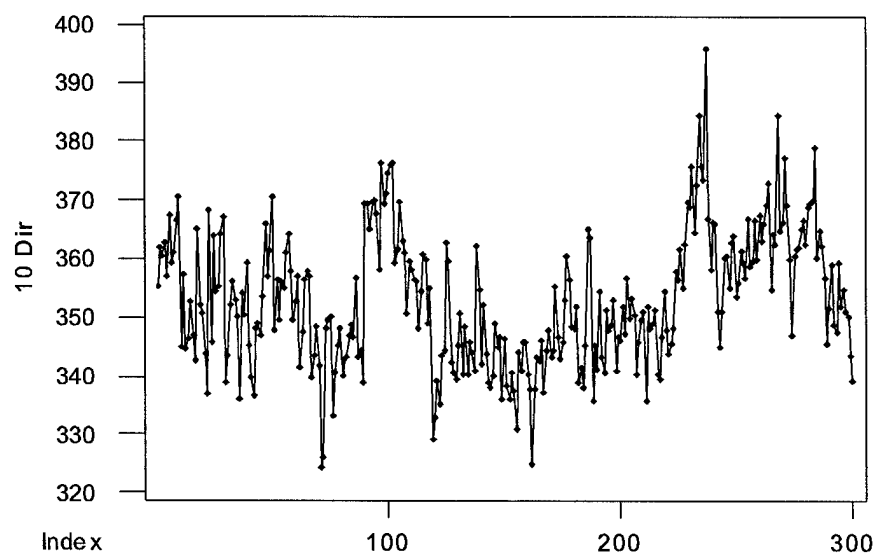


Figure 3.12: 10 second-averaged Wind Direction in Dataset st712c

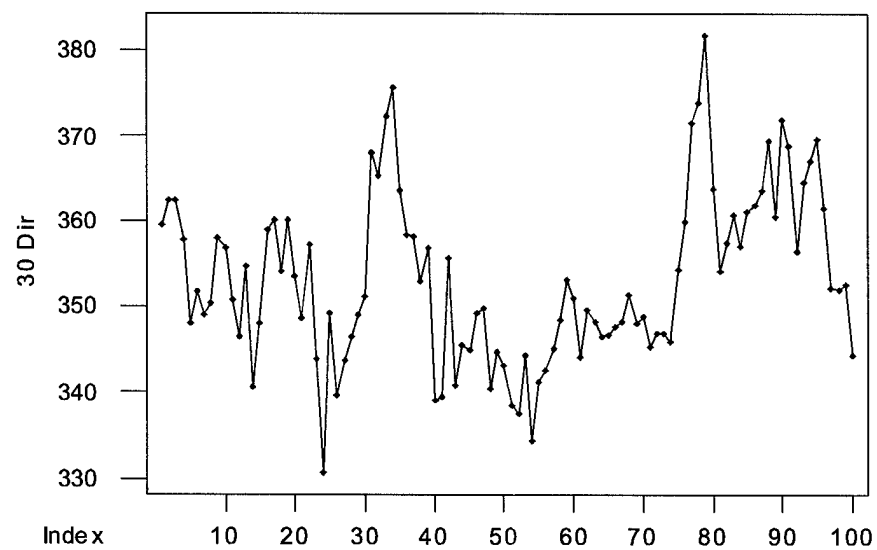


Figure 3.13: 30 second-averaged Wind Direction in Dataset st712c

Autocorrelation Function for 10 Dir

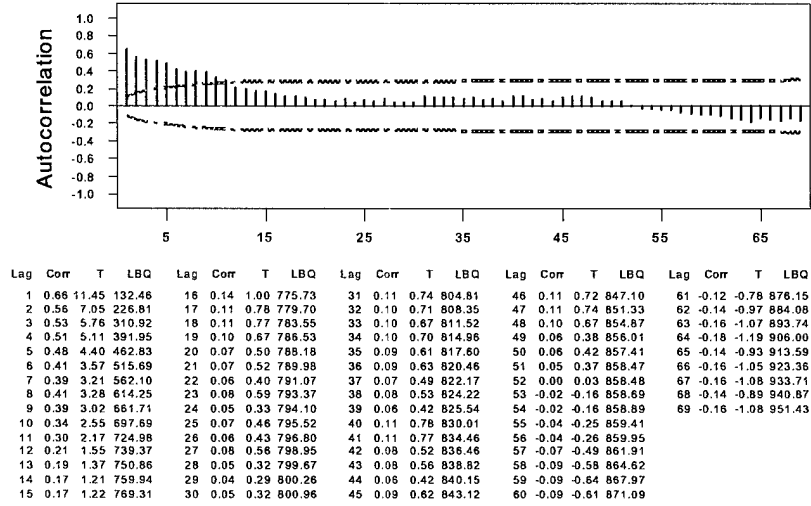


Figure 3.14: ACF for 10 second-averaged Wind Direction in Dataset st712c

Autocorrelation Function for 30 Dir

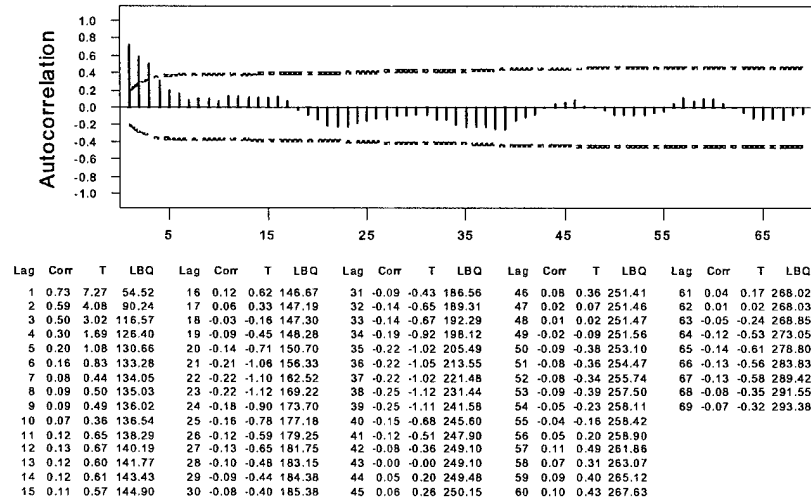


Figure 3.15: ACF for 30 second-averaged Wind Direction in Dataset st712c

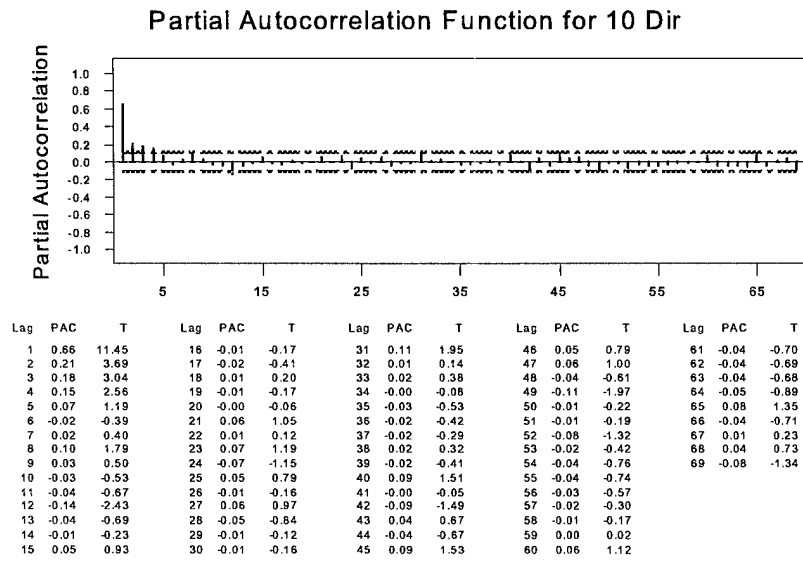


Figure 3.16: PACF for 10 second-averaged Wind Direction for Dataset st712c

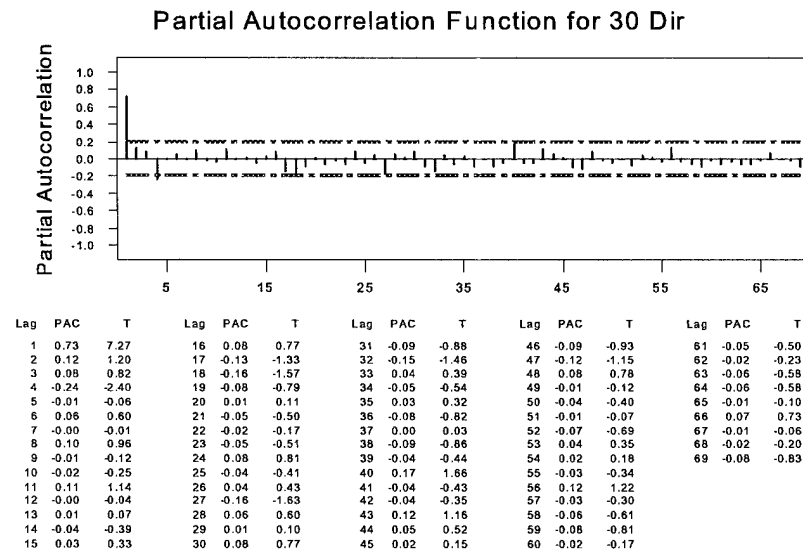


Figure 3.17: PACF for 30 second-averaged Wind Direction for Dataset st712c

and thus differs slightly from Equation 2.34. This saves the user from having to subtract the mean from the data to be modeled. For this case, if the mean is manually subtracted, the coefficient for  $z_{t-1}$  is still 0.6646, as it should be.) For the AR(1) model, the sum-squared error is 21651.2, giving a mean-squared error of 72.7 (for 298 degrees of freedom). The ACF and PACF of the residuals are significant at multiple lags, indicating correlation of the residuals and thus an unmodeled higher-order effect. The Ljung-Box statistic reported by Minitab shows significant values at lags 12, 24, and 36; this statistic is used to determine if unmodeled seasonal effects are present.

As the AR(1) model does not adequately explain the data, we proceed to add additional terms. The results of each subsequent model are shown in Table 3.1. From left to right, the columns show the order of the model, the coefficients of the model (starting with the constant at the top, the  $z_{t-1}$  coefficient, the  $z_{t-2}$  coefficient, and so on), whether the coefficient is statistically different from zero with 95% confidence, the sum-squared error for the model, the mean-squared error for the model, whether there is statistically significant correlation between the residuals, and whether the Ljung-Box test shows any statistically significant results.

From this, we see that the AR(4) model is the correct choice. With the AR(3) model, the Ljung-Box test still shows significant results. Also, the fact that all coefficients in the AR(3) model are significant supports the tentative addition of another coefficient. With the AR(4) model there is no correlation of residuals or indication of a possible unmodeled seasonal effect. Even though the second and third coefficients are not significant by themselves, the fourth coefficient is, which justifies the higher-order model. It should also be noted that Minitab does not allow lower-order coefficients to be eliminated or forced to zero.

Tentatively adding a fifth AR coefficient, we find that none of the coefficients beyond first order are significant in the AR(5) model (as a whole, the coefficients are significant, else the AR(5) model would be no better than the AR(1) model, but, individually, no coefficient “pulls its own weight”). Thus, the slight improvement of the AR(5) model does not justify

Table 3.1: Development of AR model for Dataset st712c

Model	Coefficients	Coeff Sig?	SS Error	MS Error	Res Corr?	Ljung-Box	Sig?
AR(1)	118.427	Yes	21651.2	72.7	Yes		Yes
	0.6646	Yes					
AR(2)	92.501	Yes	20642.2	69.5	Yes		Yes
	0.5218	Yes					
	0.2163	Yes					
AR(3)	75.8825	Yes	19999.5	67.6	No		Yes
	0.4834	Yes					
	0.1248	Yes					
	0.1769	Yes					
AR(4)	64.2183	Yes	19550.0	66.3	No		No
	0.4568	Yes					
	0.1061	No					
	0.1051	No					
	0.1501	Yes					
AR(5)	59.7777	Yes	19461.2	66.2	No		No
	0.4467	Yes					
	0.0989	No					
	0.0980	No					
	0.1195	No					
	0.0676	No					

its additional complexity, and it is not parsimonious.

The best model for this data set, therefore, is

$$z_t = 64.2183 + 0.4568z_{t-1} + 0.1061z_{t-2} + 0.1051z_{t-3} + 0.1501z_{t-4} + a_t \quad (3.8)$$

where  $a_t$  is a normal random variable with a standard deviation iteratively chosen so that the standard deviation of the synthetic data is 11.37.

As a sanity check, we note that, for  $z_{t-1} = z_{t-2} = \dots = 353$ , the mean value, then  $z_t = 353$ . The same is true of all of the other models.

After identification of the model, it is a simple matter to generate an instance of the random process by generating a sequence of shocks and applying Equation 3.8. Figure 3.18 shows a time-series plot of the real 712c dataset and an AR(4) simulation. Data through index 300 are the original dataset, and the rest are simulated. It must be noted that Minitab cannot be used for this method of instantiation of the process. Minitab's prediction capabilities do not include the random effects of shocks but simply operate on the tail end of the real data with the inclusion of upper and lower confidence intervals to account for the likely effects of shocks.

The ACF and PACF of the simulated data are shown in Figures 3.19 and 3.20. As a whole, the character of the functions is similar to that for the real data (Figures 3.14 and 3.16; thus, we conclude that this technique is able to realistically simulate a given dataset.

In order to restore the original sampling rate to the averaged data, the final step in the process will be to replicate each point to restore the original rate, and a small amount of random noise will then be added to each point.

Although the wind speed can also be modeled as an ARIMA process (requiring, perhaps, a different averaging time or order of model), for the current method does this is probably not necessary as the wind speeds are uncorrelated using a ten second averaging time. Thus, we simply model the wind speed as a normal random variable with the desired mean and standard deviation.

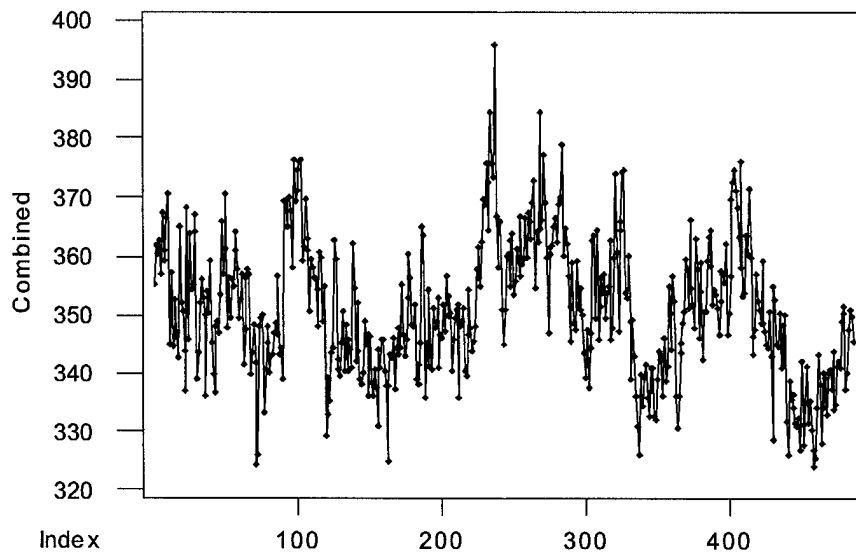


Figure 3.18: Real and Simulated Direction Data for Dataset st712c

#### Dataset st712d

After demonstrating that a time-series model can be developed for a single experimental data set, we analyze other data to determine what characteristics of the models are similar. We next analyze data set st712d. The set was taken shortly after set st712c, but is class D due to calming conditions in the evening. Any small portion of st712d looks similar to st712c but the trends in the data as a whole are different; whereas st712c has fluctuations of various scales about its mean, st712d seems to have a dominant periodic movement.

Although there are high-lag points at which the PACF is significant for 10, 15, and 30 second averaging, the PACF for 10-second averaging cuts off after roughly lag 5. We will proceed with the 10-second averaging for easy comparison with the previous example. The dataset for this case is shown in Figure 3.21, and the PACF in Figure 3.22.

Autocorrelation Function for Sim 10 Dir A

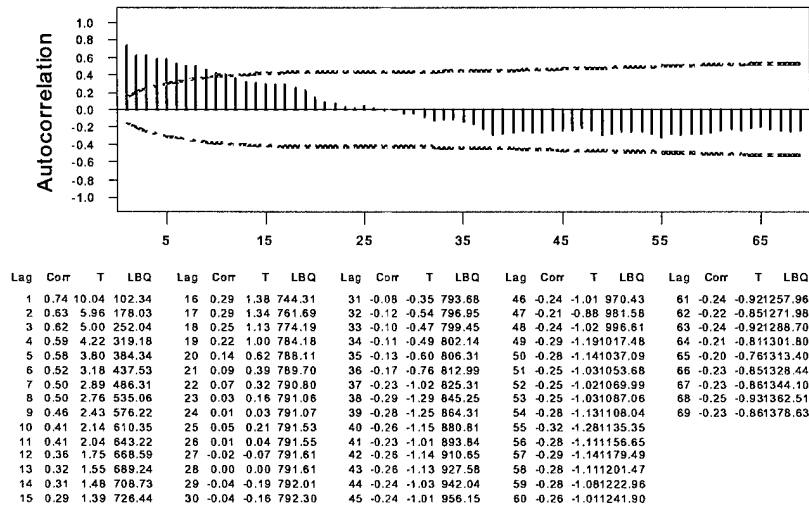


Figure 3.19: ACF for Simulated 10 second-averaged Wind Direction for Dataset st712c

Partial Autocorrelation Function for Sim 10 Dir A

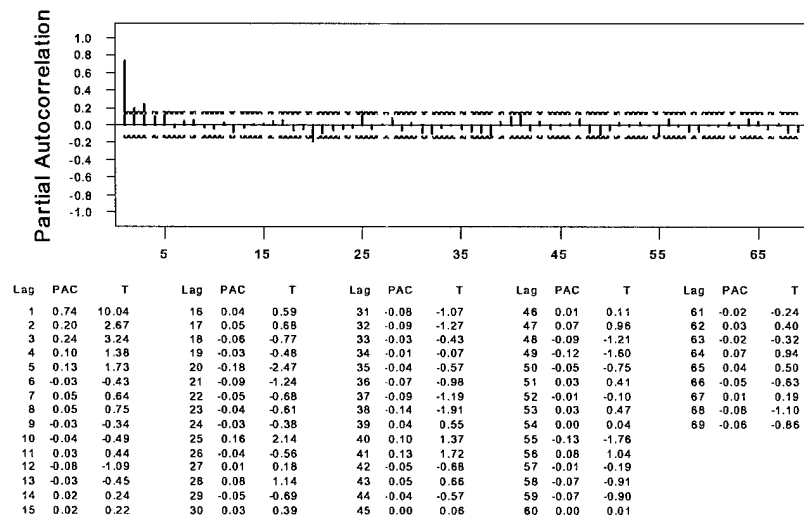


Figure 3.20: PACF for Simulated 10 second-averaged Wind Direction for Dataset st712c

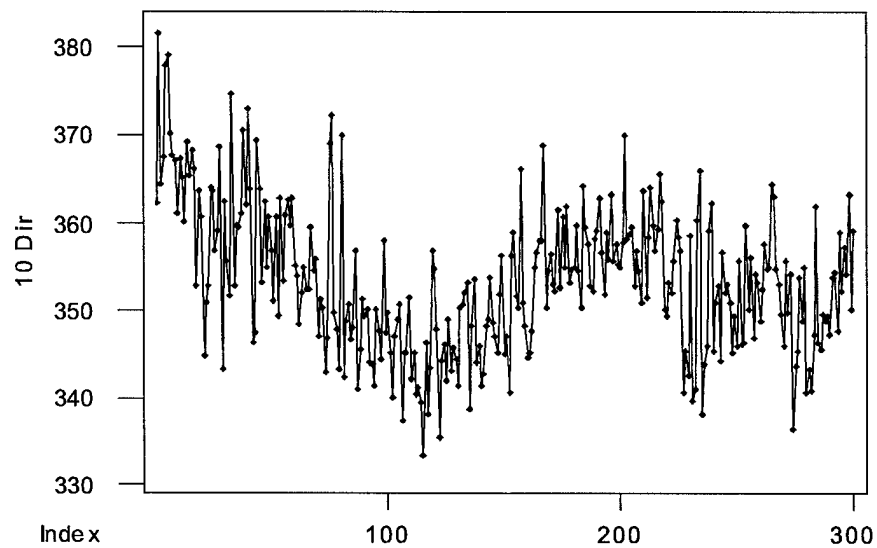


Figure 3.21: 10 second-averaged Wind Direction for Dataset st712d

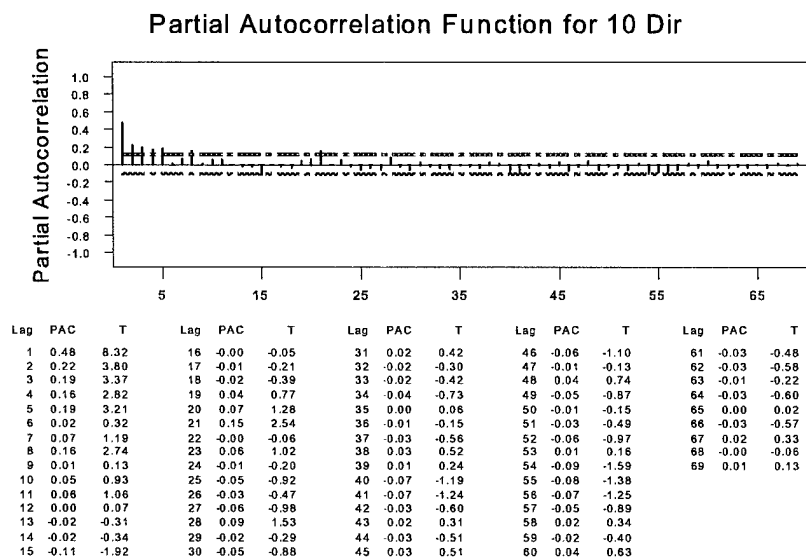


Figure 3.22: PACF for 10 second-averaged Wind Direction for Dataset st712d

Developing an ARIMA model for this dataset is problematic. Figure 3.21 shows that the data meander in a long-term fashion, which likely violates stationarity. Indeed, though all coefficients are very significant for an AR(4) model, the fourth-order model also shows significant Box-Ljung results, indicating that the non-seasonal model is inappropriate. Adding a fifth coefficient, Minitab reports that the back forecasts do not die down, indicating that the data are not stationary. Although the model is not adequate, simulated results are shown in Figure 3.23 to illustrate the time-domain manifestations of these difficulties. The actual data set runs through lag 300, followed by simulated data.

Unlike the previous dataset, there is an obvious difference between the real and simulated data for this case. The real data exhibit a long-period meander that is not included in the simulated data. The simulated data fluctuate more from one value to the next to capture the overall spread of the actual data, but do not do so in the proper systematic way.

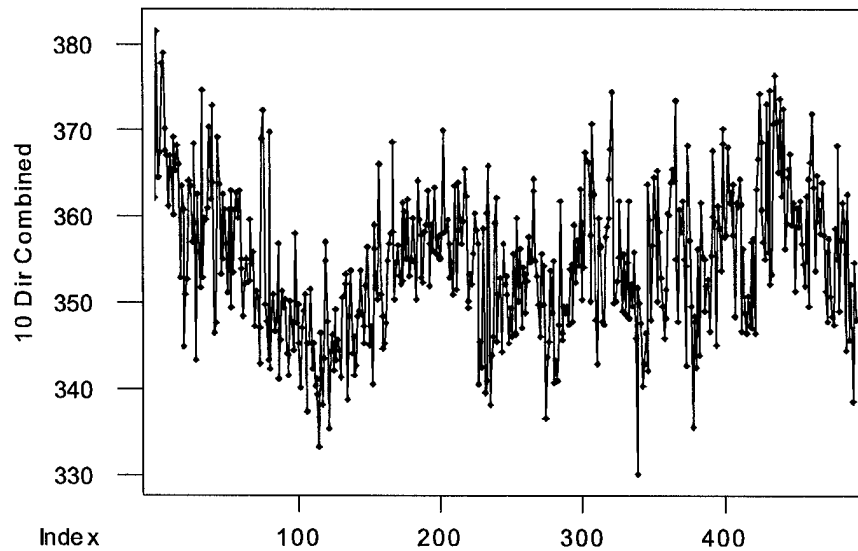


Figure 3.23: Real and Simulated Direction Data for Dataset st712d

A second model was next developed to account for the meander in this data. First, a sinusoid was fitted to the original data and subtracted from it; the fit is shown in Figure 3.24.

The residual data were then analyzed in Minitab. The best model found was an AR(5) model. Only the first and fifth coefficients of the model are actually significant (though all must be retained). The Box-Ljung statistic is not significant, indicating that the non-seasonal model is adequate. However, the correlation of the residuals is significant at low lags; a primary cause of this is that the experimental values are predominantly higher than the fit values at both ends, as shown in Figure 3.25.

Despite its drawbacks, this model was used to generate synthetic data, in this case of the residual, to which the original sinusoid was added. The results of this simulation are shown in Figure 3.26. Again, the real data run through index 300, followed by the synthetic data. There is a slight jump at the start of the synthetic data because the starting phase of the sinusoid that was added to the synthetic data was not matched to the ending phase

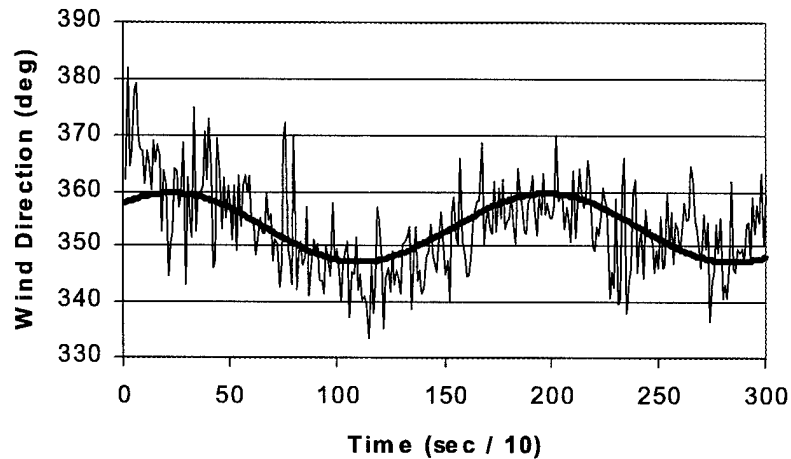


Figure 3.24: Direction Data for Dataset st712d, with Sinusoidal Fit

of the real data.

The synthetic data meander in a manner similar to the real data, though the typical point-to-point variation still seems a bit higher than in the real data. This is an effect of the “smiling” residual curve; the systematic difference between the data and the sinusoid increases the standard deviation of the difference data, but in a systematic way; when the data are simulated, this extra difference becomes part of the random point-to-point fluctuation.

To attempt to correct this deficiency, a third model was developed. Rather than using a sinusoidal fit, this model uses a ten-point moving average smoothing as a fit to the data, as shown in Figure 3.27. The difference between the value at each point and the moving average was analyzed in Minitab.

After subtracting the moving average from the data, it was found that the residuals (between the experimental data and the moving average) were uncorrelated. Thus, no ARIMA model is necessary, and the dataset (at least with 10 second averaging) is adequately

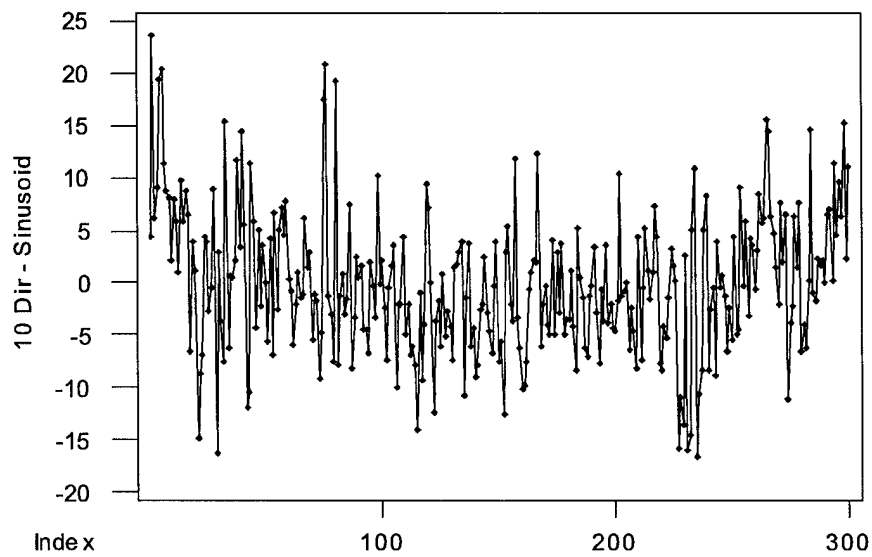


Figure 3.25: Residual Error between Dataset st712d and Sinusoidal Fit

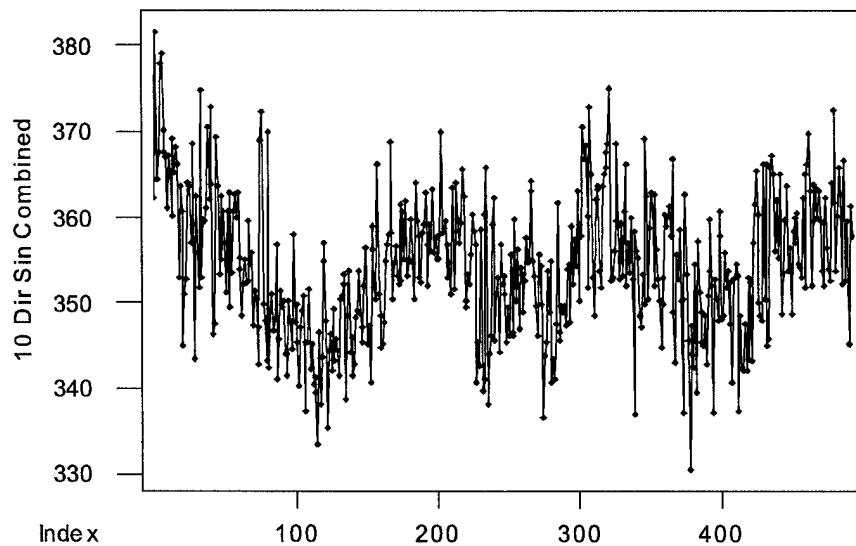


Figure 3.26: Real and Simulated Direction Data for Dataset st712d, with Sinusoidal Baseline

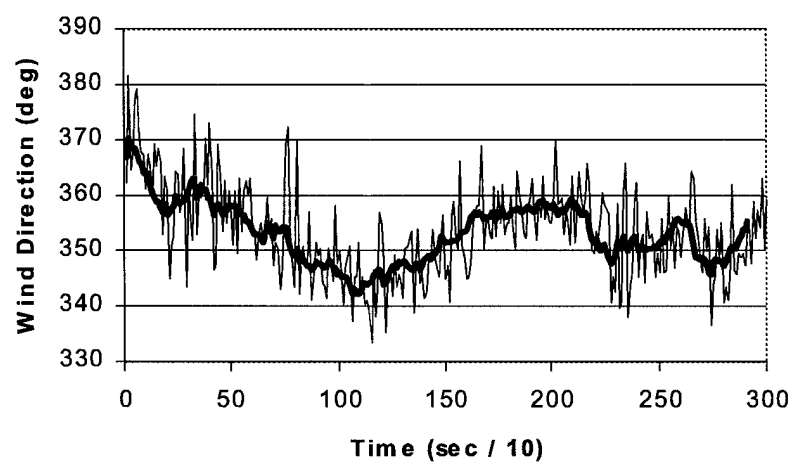


Figure 3.27: Direction Data for Dataset st712d, with Moving Average Fit

described by a baseline onto which Gaussian noise is superimposed.

For reconstruction, an appropriate baseline must be used. As the moving average baseline from the experimental data is not easily characterized nor necessarily intrinsic to similar data, we instead use a sinusoidal baseline obtained by fitting a sinusoid to the moving average baseline; the resulting sinusoid (shown in Figure 3.28) differs very little from the sinusoid fit to the actual data (Figure 3.24). The final model is given by

$$z_t = 6.166818 \sin(0.035869t + 0.82983) + 353.0602 + (a_t + 0.264) \quad (3.9)$$

where  $a_t$  is Gaussian noise chosen so that the standard deviation of  $z_t$  is 5.976. The Gaussian noise also has 0.264 added to it, which is the mean of the residual between the data and the moving average base.

The results of a simulation with this method are shown in Figure 3.29 (again, with real data through point 300). This simulation seems to capture the characteristics of the experimental data better than the other two models.

### **Dataset st712e**

Finally, the third dataset from July 12, 1996 is analyzed. Taken shortly after st712d, this set is also estimated to be stability class D. From examination of the ACFs and PACFs, it is found that, again, 10-second averaging is appropriate. The 10-second averaged data are shown in Figure 3.30.

At first glance, there is no overall long-term meander to this data, but there is a temporal change in behavior. At the beginning of the dataset the direction oscillates with a small amplitude, with a meandering centerline. After approximately 120 data points the data begin to oscillate more, and the local centerline of the oscillation moves more randomly.

Following a trial-and-error approach, an AR model was developed for the data without any prior treatment to determine whether the variations cause difficulties with model development. An AR(4) model was found to be satisfactory. The first and fourth coefficients were significant, residuals were uncorrelated, and the Box-Ljung test did not show

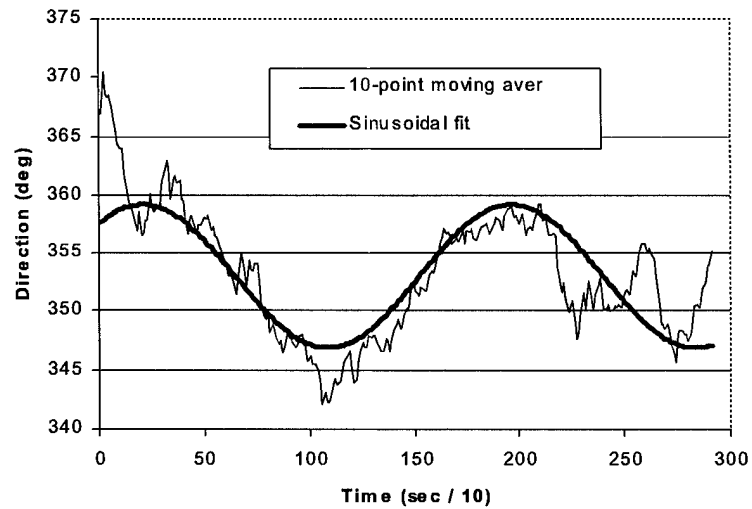


Figure 3.28: Sinusoidal Fit to Moving Average Baseline for Dataset st712d

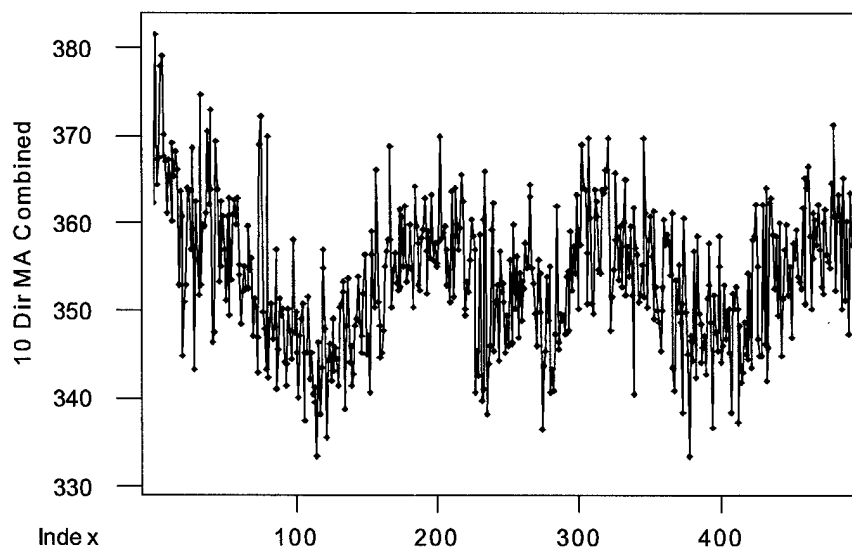


Figure 3.29: Real and Simulated Direction data for Dataset st712d, Gaussian Noise on Sinusoidal Baseline

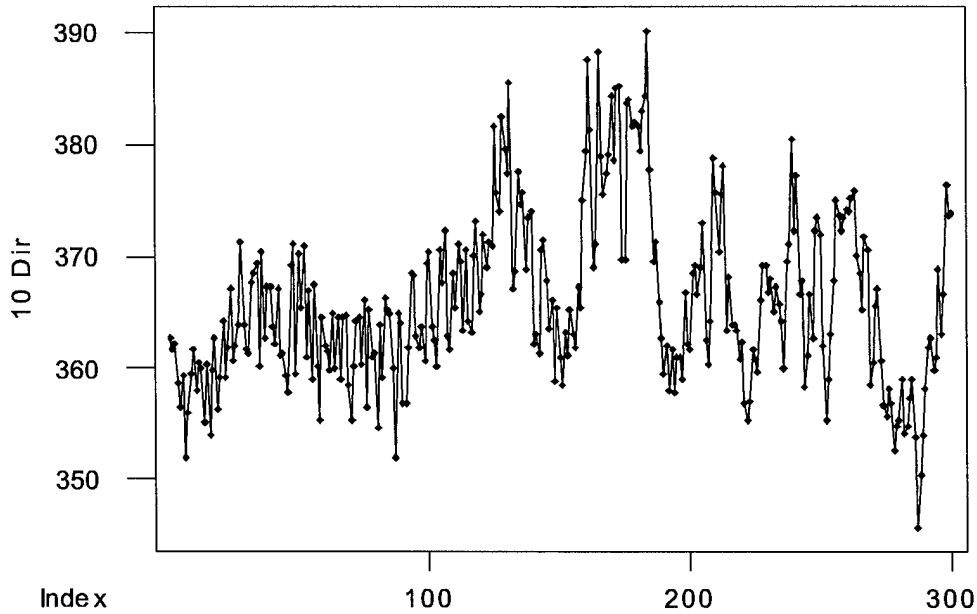


Figure 3.30: 10 second-averaged Wind Direction for Dataset st712e

significant results. Thus, this AR(4) model seems satisfactory for the data. The model is

$$z_t = 56.3502 + 0.5487z_{t-1} + 0.0807z_{t-2} + 0.0847z_{t-3} + 0.1321z_{t-4} + a_t \quad (3.10)$$

where  $a_t$  is Gaussian random noise with a standard deviation chosen so that the standard deviation of  $z_t$  is 7.67, the standard deviation of the experimental data.

Results using this model are shown in Figure 3.31. In general, the simulated data (after point 300) matches the character of the second portion of the experimental dataset well.

### 3.2.4 Guidelines for Development of Additional Wind Models

We have developed models for three experimental data sets in this section. Unfortunately, the final models are insufficient to allow the user to infer any rules for general model parameters for similar data sets. We suggest two areas of focus for improving understanding of wind behavior. First, common conditions should be studied in detail to determine typical

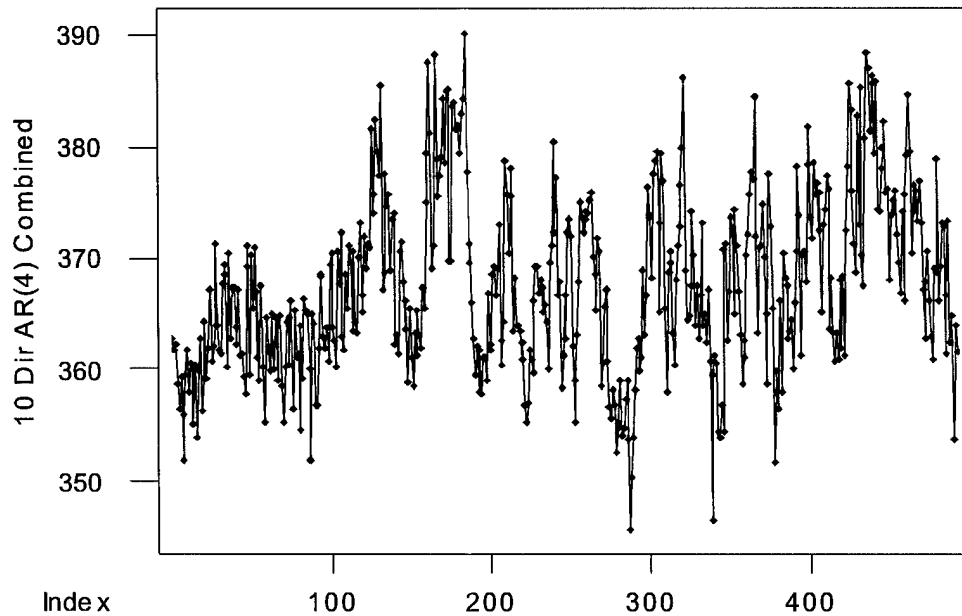


Figure 3.31: Real and Simulated Direction Data for Dataset st712e

parameter values with high confidence. Second, a variety of conditions should be studied to determine general variation of parameters with conditions.

#### **Development of general parameters for basic conditions**

A fundamental step in developing a comprehensive understanding of wind model parameters is the collection and analysis of additional wind data sets for conditions similar to our existing data (classes C and D). Ideally, some data should be collected at high frequencies (about 1 Hz). The autocorrelations of these data sets should then be analyzed to determine a meaningful time scale at which the data sets are not overcorrelated. If this time scale is on the order of 10 seconds, as it was for our data sets, this would be a significant general property. Once the time scale is determined, additional data collected at the significant frequency should be collected.

After data are collected, AR models should be developed for each data set. A simple

procedure for doing this using the R (R Development Core Team, 2001) software is in Appendix E. Once multiple models for similar conditions are available, bounds on parameters can be drawn. These bounds could then be used to randomly generate AR coefficients given a set of conditions.

If available equipment is not capable of recording high frequency data, drawing general conclusions would be more difficult. However, it would still be possible to compare whatever new data are available to averaged values for existing high-frequency data sets. This would hopefully allow at least a qualitative assessment of whether the existing high-frequency data are typical for certain conditions.

### **Development of models for less-typical conditions**

In addition to comprehensively developing parameter ranges for “benign” conditions, models for less ideal conditions should also be developed. Eventually, these could be studied to the same level of detail as classes C and D, but preliminary study should focus on a collecting data for a wide sampling of conditions (time of day, insolation level, mean wind speed, temperature, etc.). One or two models could be developed for each set of conditions. The goal of this portion of the study would be to determine in general how models for, for example, a gusty, unstable night compare to models for neutral nights or days in terms of appropriate model order, significance of coefficients, and magnitude of coefficients.

It should be possible to draw some conclusions for a variety of conditions with low-frequency wind data if high-frequency data are not available.

### **3.2.5 Default Model Parameters**

Appendix C describes the parameters used in the CALPUFF GPI code. They are divided into functional areas, and the names of the variables and input groups and their units are given.

### 3.3 Roof Vent Modeling

Integration of the roof vent results into DIRSIG is a bit more complex conceptually than integration of results for transmissive/emissive plumes. Absorption and emission by plumes are linear effects, and their impact on a ray can thus be easily superimposed with other effects. Temperature and heat transfer effects, on the other hand, cannot simply be simulated by superposition of effects. For example, the heating of the surround by a plume tends to increase its temperature; the temperature increase does not change the solar heat loading, but it changes the rates of heat transfer due to radiation and conduction (and convection should the properties of the convecting medium change). The end result is that, when a new heating effect is introduced, the thermodynamic equilibrium of a body is changed in a nonlinear way.

Thus, treating heat transfer in a fully consistent manner requires a comprehensive and expandable thermal model. As this is not available in DIRSIG, we must for the time being use existing capabilities. In this case, we have chosen to use the existing capability of specifying wind temperature and speed that is used to calculate convective transfer to surfaces.

We ignore the nonlinear nature of multiple effects in two ways. First, the plume model calculates a temperature difference between the plume and the ambient temperature at a point. This is then added to the air temperature already calculated by the DIRSIG thermal model. This superposition effectively can be viewed as assuming that air from the vent does not displace existing air in any way; instead, the vent's heat is transported by some mediumless method, whereupon it warms the existing air without disturbing it. We have found that the alternative, simply using the vent's temperature as the existing air temperature, tends to cause the thermal model's temperature equilibrium iteration to become unstable. Simply using the vent temperature would also tend to make the air temperature too uniform; in reality, mixing will tend to cause the temperature to be higher

in regions where it would be higher in isolation.

Furthermore, we have no convenient method for treating areas with heating effects from multiple vents. We thus have chosen to allow each point of interest to be affected only by the vent that, in isolation, would produce the highest temperature at that point.

### 3.3.1 Transverse Vents—Halitsky GPI Client

As discussed in Subsection 2.5.4, transverse vents are not particularly significant for SIG applications. We have implemented Halitsky's plume model mainly as another example client. It might be useful to modify it to replace the CALPUFF GPI for direct sensing of small stacks (although this re-introduces the ray tracing difficulties solved by using a puff-based approach). The jet region of the model could be added in (we do not consider it since heated plumes do not re-contact the roof until further downstream) and used for simulation of a LADAR or other velocity measuring device.

The flow for this client is much simpler than for the CALPUFF GPI because there are no plume parameters requiring or benefiting from pre-computation. The only task of the `gpi_init` routine is to call the standard GPI routine to read the release file. All plume calculations are done in the `gpi_point` routine as shown in Figure 3.32.

We first assume that the plume has progressed beyond the jet region, so that the wind speed everywhere is the same as the freestream wind speed. Given a point of interest on the roof or ground, it is then simple to calculate the downwind and crosswind locations of the point by projecting the  $x$  and  $y$  coordinates onto the downwind and crosswind directions. Then, the rise is calculated exactly as it is for factory stacks through Equations 2.17, 2.18, and 2.20. The dispersions are found from Equations 2.44 and 2.45 (as shown in Figure 2.22). From these, the dilution is found from Equation 2.46; the temperature is then found in the same manner as for factory stacks. The wind speed is also reported back to DIRSIG, but, as we assume there is a temperature elevation on the roof only in the main region, the wind speed returned is always the freestream wind speed.

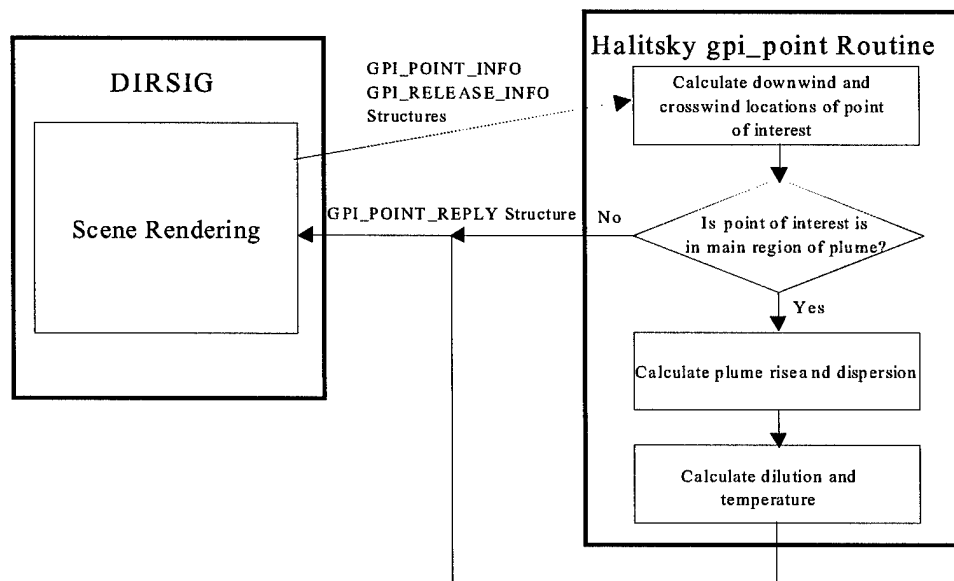


Figure 3.32: Point Calculation Flow for Halitsky GPI Client

### 3.3.2 Axial Vents—Abramovich GPI Client

The problem for axial vents is similar to that for transverse vents. Again, the `gpi_init` routine only requests that the release file be read. All processing is again done in the `gpi_point` routine.

The basic basic flow (Figure 3.33) is very similar to that for the `halitsky_gpi` client; the differences are in the actual equations used to calculate the temperature and velocity. The primary difference between the two procedures is that concentration is the fundamental variable for the Halitsky model, from which temperature is derived (with no wind speed variation), whereas centerline wind speed is the fundamental variable for the Abramovich model, from which temperature and off-centerline properties are derived. Additionally, the calculations for the Halitsky model are explicit, whereas the Abramovich model iterates on wind speed until the desired downwind distance is matched.

Given a point of interest, we calculate the downwind and crosswind distances. From

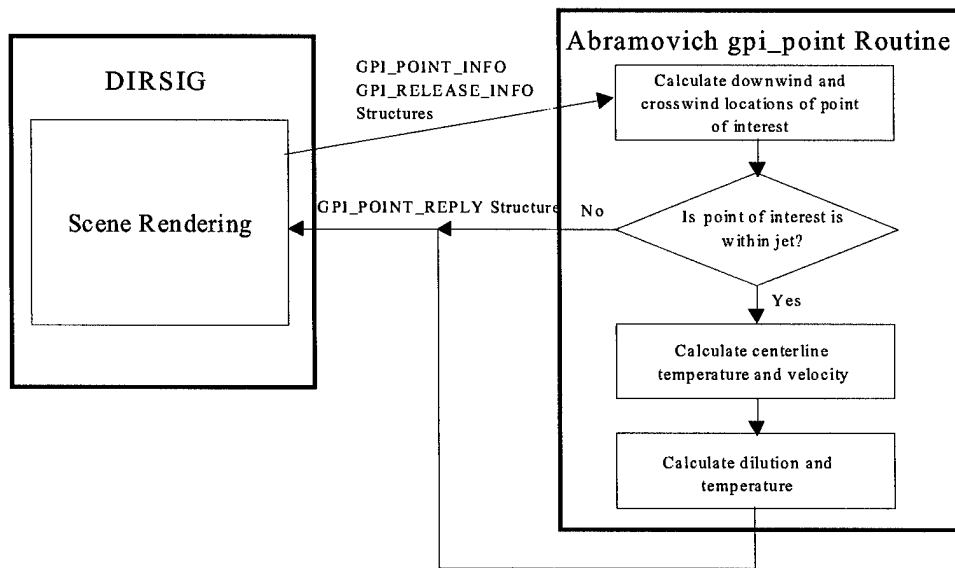


Figure 3.33: Point Calculation Flow for Abramovich GPI Client

the downwind distance, the centerline velocity is found using the procedure outlined in Appendix D. The dispersion is calculated using Halitsky's curve fit (Equation 2.44). With the dispersion, we can then find the velocity at the point of interest. We assume that the temperature dilutes in the same manner as velocity (as discussed in Subsection 2.5.8), from which we calculate the temperature. The temperature and wind speed are reported back to DIRSIG in the same manner as for transverse vents.

### 3.3.3 Downward Vents—`cf_d_gpi` Client

Downward-discharging vents are in some ways both the most complex and simplest case to implement in DIRSIG. The complexity arises from the need to calculate the effects of such vents using Computation Fluid Dynamics (CFD). Before running DIRSIG, the user must run the CFD code and extract the results. The process of running the CFD code is involved and requires some knowledge of fluid mechanics.

Preparing input to the CFD code requires several steps. First, the user must define in detail the geometry of the problem including vent location, the wall boundary, and free flow boundaries. Second, the solution volume must be discretized into a computational grid on which flow properties are calculated. Finally, boundary conditions on the grid boundaries are defined. These conditions simulate walls and freestream conditions.

Once the flow geometry is created, execution of the code is time consuming; run times on the order of fifteen hours, even when the problem is executed in parallel on multiple machines, are not uncommon. Once the code is run, the user must check the reasonableness of the results and ensure that the solution is converged. Finally, if flow conditions are changed, the entire process must be repeated as the solution is strictly valid only for the input conditions.

Once the flow field has been computed, however, incorporation of the results into DIRSIG is nearly trivial. The `gpi_init` reads the CFD data in to memory, where it is held until it is needed during ray tracing. The `gpi_point` portion of the client simply receives a point of interest and determines the temperature and velocity by interpolating on the CFD solution grid.

The procedures for running the CFD solution will vary with the code used. We have chosen to use the WIND code, developed by NASA and the U.S. Air Force. This code is available to U.S. entities possessing export licenses, and is thus available to nearly all users of MODTRAN and DIRSIG. An overview of the specific process used to define a problem, run WIND, and extract the results can be found in Appendix F. This guide is not intended to be a complete tutorial. The user should first work through the *case4* example in WIND User's manual (NPARC Alliance, 2001). This example includes essentially the same model generation and execution steps as our CFD model. Once the *case4* example is understood the procedures in Appendix F detail the specific tools and files developed for our basic roof vent.

If the user does not wish to develop a new solution, our existing solution may be usable.

Fortunately, the flow parameters for our downward vent are also reasonable for exhaust from a large truck. Also, it is reasonable to scale the exhaust temperature by some small amount from the value used in the CFD solution. The temperature at a point of interest can be approximated by

$$T = T_a + (T_0 - T_a) \frac{T_e - T_a}{T_{e0} - T_a} \quad (3.11)$$

where  $T_a$  is the ambient temperature,  $T_e$  is desired exhaust temperature,  $T_{e0}$  is the exhaust temperature used in the CFD solution, and  $T_0$  is the temperature at the point of interest from the CFD solution.

## Chapter 4

# Results

### 4.1 CALPUFF Client Validation

Before using it for generation of imagery, we first validate the CALPUFF client to the extent possible. Although we have little actual imagery to directly compare to our results, we can compare our results to other researchers' results and appeal to their validation.

#### 4.1.1 Heritage from MIND Validation Results

##### Concentration Time Series Simulation

As discussed previously, we use two parts of the MIND algorithm. First, we smooth the wind, using the smoothed wind direction to locate the plume centerline. Second, we use the smoothed wind to generate instantaneous diffusions. Both of these portions of the algorithm produce results identical to MIND when given identical inputs (experimental wind data acquired by Dr. Peterson and her colleagues). As the MIND code itself assumes neutrally buoyant effluent discharged at ground level, we cannot directly compare the numerical results further. Furthermore, MIND intrinsically uses angular dispersion coefficients rather than linear coefficients so that its concentration measurements do not match ours exactly (although they are close, assuming the angular dispersions are small angles).

Nevertheless, as the results tend to be very close, we appeal to some of MIND's validation results. All figures referenced in this section are from O'Neill (1996b). First, we show observed and predicted concentrations for a location near the plume centerline for tests st712c and st712d in Figures 4.1 and 4.2. (O'Neill (1996a) shows a similar figure for st712e, but the plume impinged on the receptor for only a small fraction of the test due to a shift in the mean wind direction; the period of impingement and peak concentration were predicted quite well by MIND). Concentrations are reported in parts-per-trillion (ppt). The receptor was located at a downwind distance of 200 m.

In general, the locations and directions of fluctuations for both observed datasets are predicted reasonably by MIND. The relative amplitudes are not always correct, and both predicted series have higher average concentrations level than the observed series. As the bias is systematic, it is likely due to unmodeled effects rather than being a random error. The data in (O'Neill, 1996b) show that the bias varies from about 0.25 to 0.7 standard errors.

### **Average Concentration Simulation**

The averaged MIND results (for 50 minutes) as compared to Pasquill-Gifford results for datasets st712c, st712d, and st712e are shown in the following figures.

These results compare quite well; the st712c dataset, identified as Class C, lies quite close to the Pasquill-Gifford Class C curve. The st712d dataset, taken just after st712c, is tentatively identified as Class D, but is in fact something of a borderline case, occurring at dusk. The result for st712d lies between the Class C and Class D curves. Finally, the st712e dataset agrees very well with the Class D curve. The predicted mean concentrations tend to be higher than the measured mean concentrations when the measurement points are well within the plume. This is likely due to the same systematic bias as seen in the discussion of time series comparisons in the previous subsection. As shown in O'Neill (1996b), the predicted peak-to-mean ratios agree very well with the observed ratios; since the ratios are

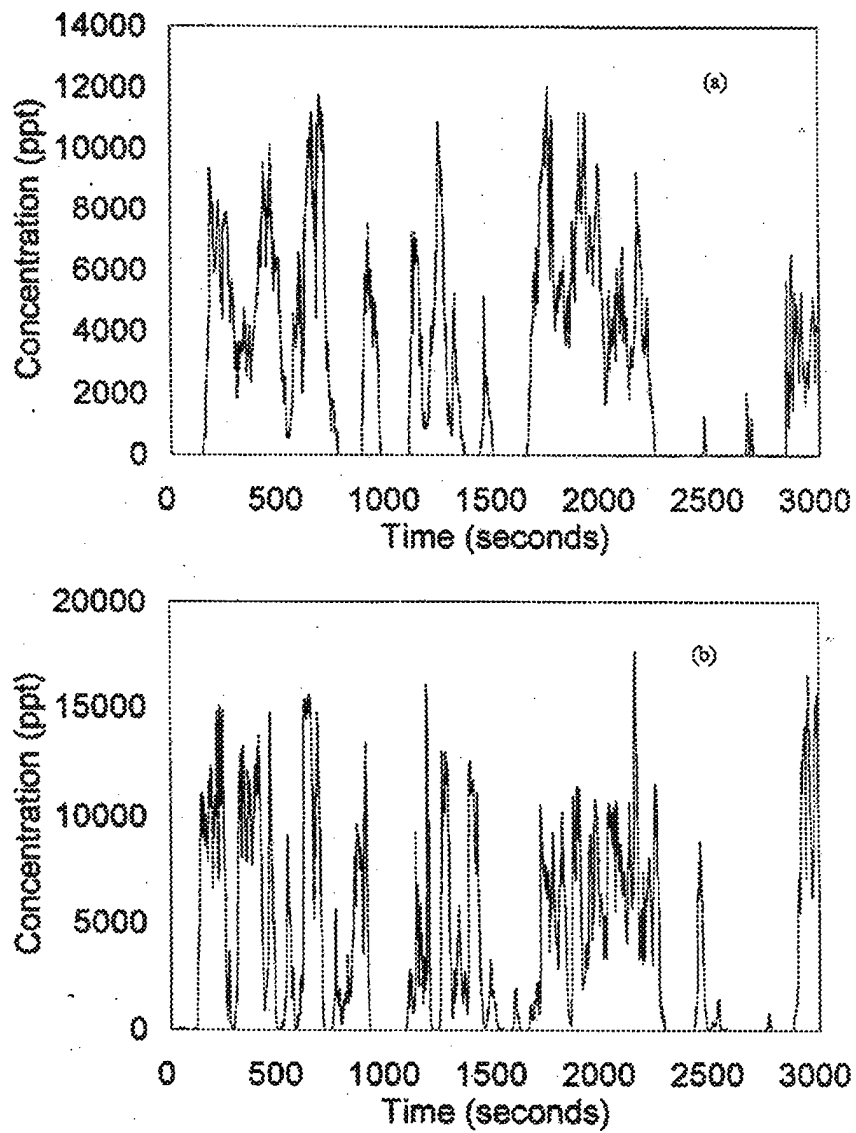


Figure 4.1: Time Series of Predicted (top) and Observed (bottom) Concentrations for Test st712c (from O'Neill (1996b))

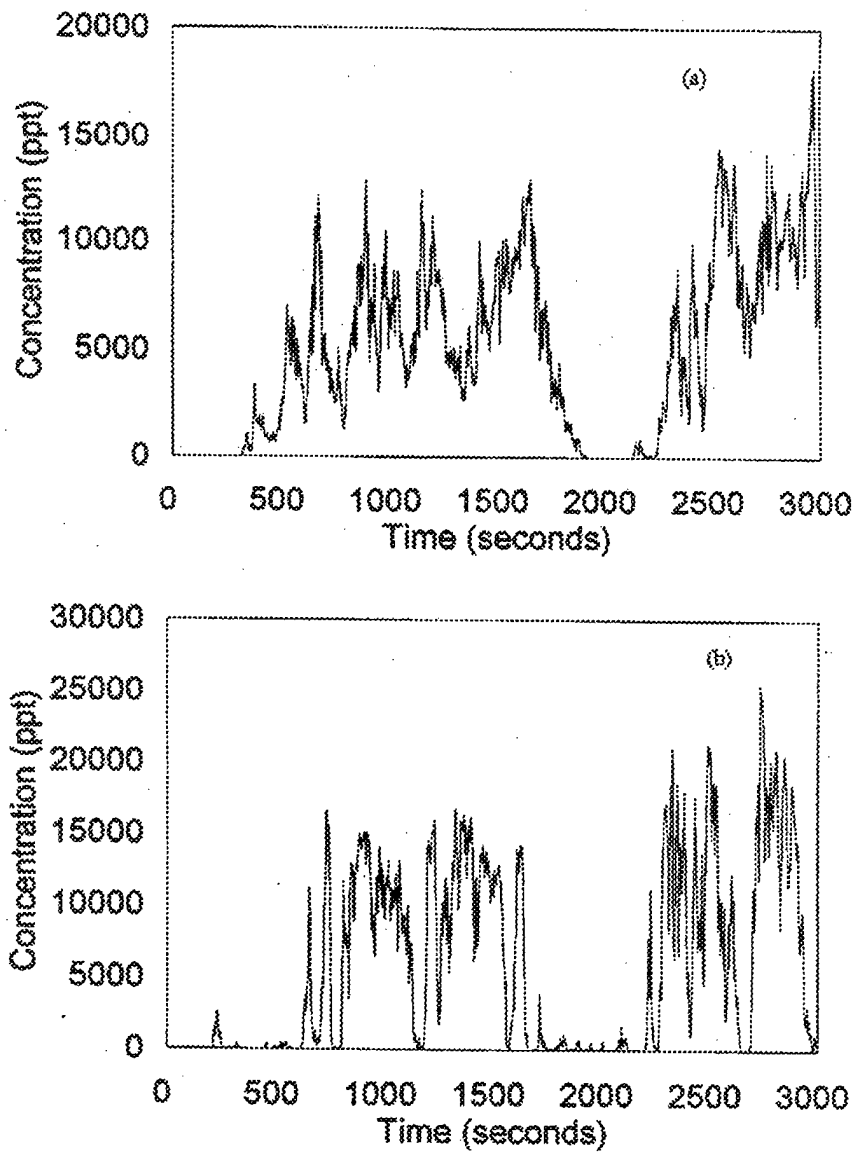


Figure 4.2: Time Series of Predicted (top) and Observed (bottom) Concentrations for Test st712d (from O'Neill (1996b))

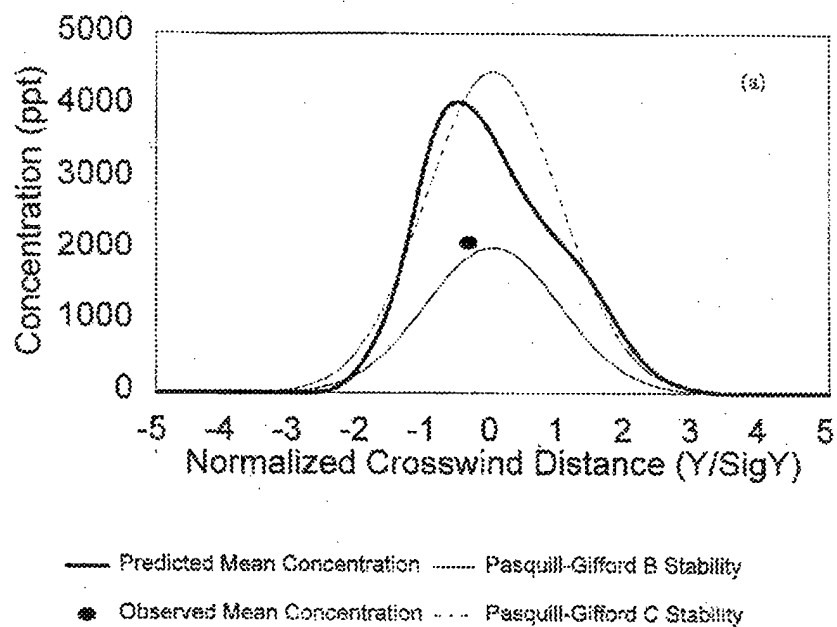


Figure 4.3: Average concentration for test st712c (from O'Neill (1996b))

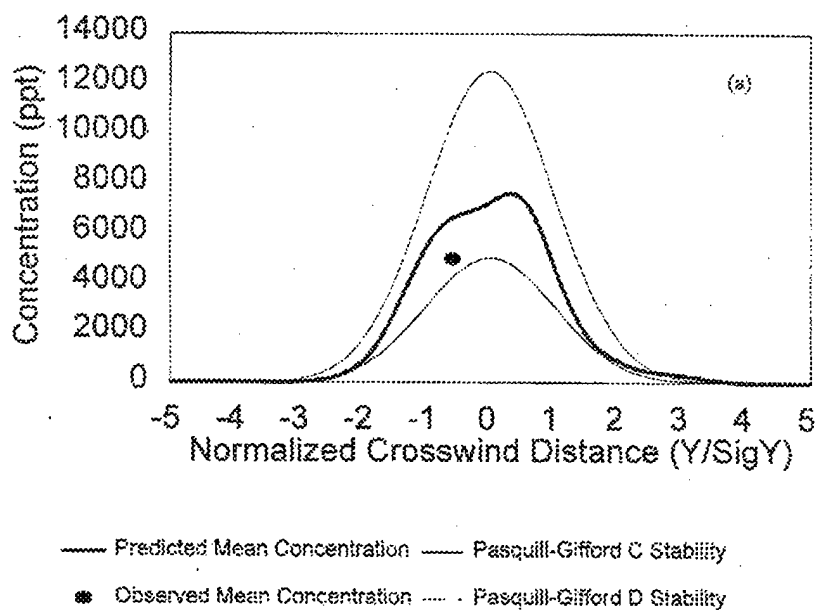


Figure 4.4: Average Concentration for Test st712d (from O'Neill (1996b))

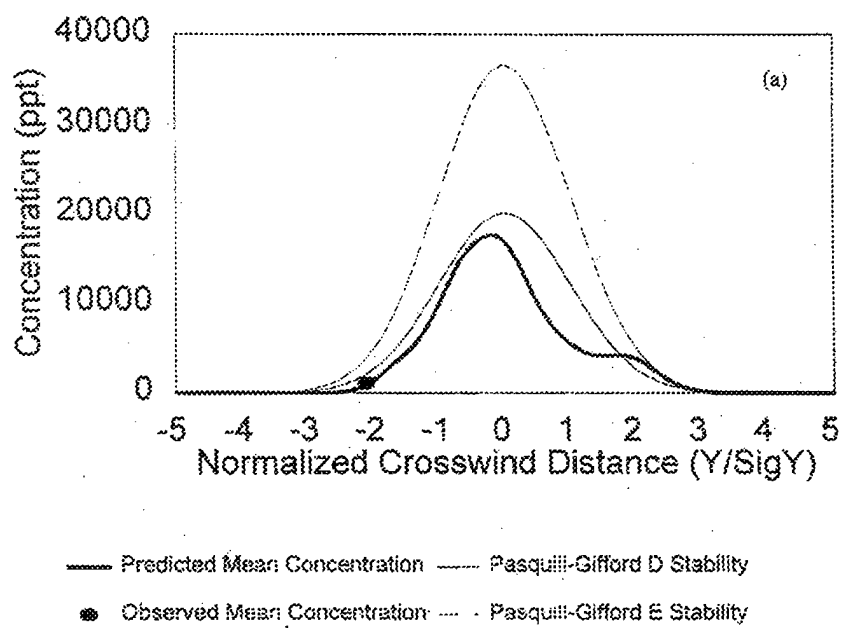


Figure 4.5: Average Concentration for Test st712e (from O'Neill (1996b))

the same, the systematic error observed in the time sequence plots (of which an error in peak is a consequence) is also seen in the averaged crosswind profiles (a measure of mean).

#### 4.1.2 Near-field Time Average Validation

##### st712c Dataset, 70 meters Downwind

In the absence of field data, it is not possible to directly compare CALPUFF GPI results to observed concentration time histories; however, we can compute averages and compare these to Pasquill-Gifford curves. For our calculations, we actually trace a horizontal ray near the plume and perpendicular to the mean wind direction once per second for a specific length of time. The concentrations along this ray are accumulated and averaged.

As we are typically interested in imaging plumes close to the stack, we use MIND's 200 m data as an outer bound and work inward. Beginning at a distance of 70 m, we trace across plumes using simulated wind data for dataset 712c for averaging times of 60, 300, 600, and 1200 seconds (plotted in Figures 4.6 through 4.9). For two reasons we do not modify the Pasquill-Gifford curves to approximate the effect of decreased averaging time, even though CALPUFF allows for a simple power law modification. First, O'Neill (1996b) does not make this modification; comparison between our conditions and O'Neill's is therefore easier if no correction is made. Second, as was stated in Hanna et al. (1982), Pasquill's curves should represent average dispersions over a few minutes; as our averaging times are almost all ten minutes or shorter, we lack strong motivation to modify the default curves.

Several observations should be made about these figures. First, the curves for short integrating times are not very Gaussian. The plume undergoes rapid motion during the first minute for this dataset (as can be seen in the NTS video) causes the average to smear and be flat. As the integrating time increases, a more central distribution develops—the rapid motion tends to be averaged out.

Second, the curves for an averaging time of 1200 seconds (Figure 4.9) is very similar to the curve for 600 seconds (Figure 4.8). This indicates that the plume essentially repeats

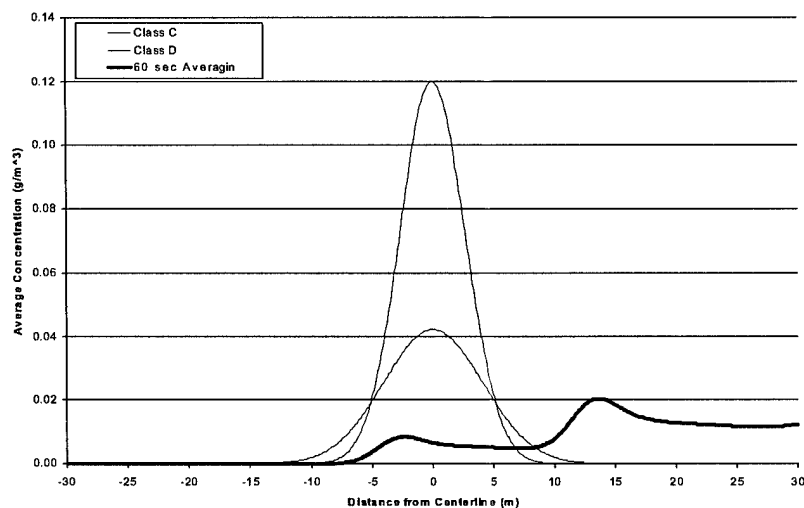


Figure 4.6: Average Concentration over 60 seconds for Test st712c at 70 meters Downwind

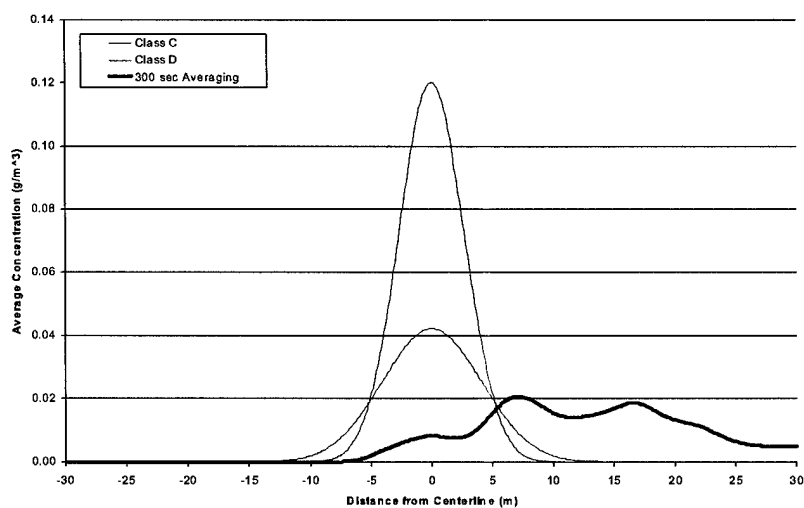


Figure 4.7: Average Concentration over 300 seconds for Test st712c at 70 meters Downwind

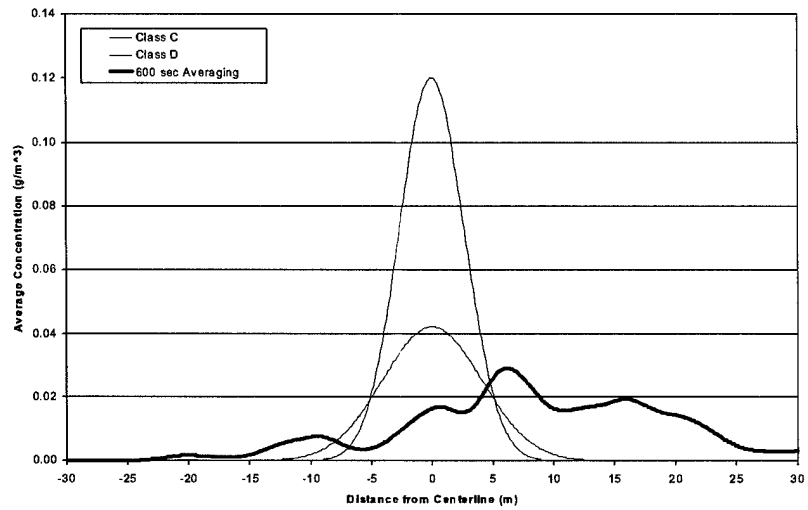


Figure 4.8: Average Concentration over 600 seconds for Test st712c at 70 meters Downwind

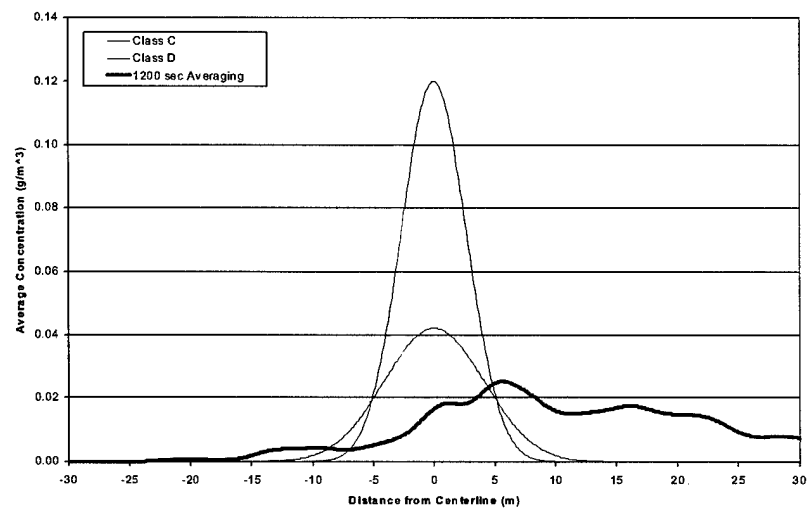


Figure 4.9: Average Concentration over 1200 seconds for Test st712c at 70 meters Downwind

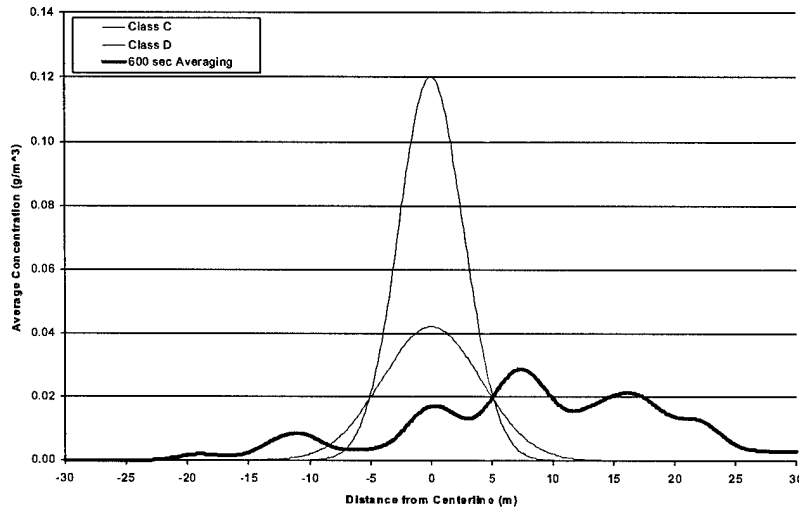


Figure 4.10: Average Concentration over 600 seconds for Test st712e at 70 meters Downwind

its motion after 600 seconds, giving the same average behavior.

In general, the comparison somewhat validates the CALPUFF GPI model. The averaged results, though not strictly Gaussian, have half-widths on the same order as the proper Gaussians. We note that the area under our curves may be more than that under the Pasquill-Gifford Gaussians. This is to be expected; the Pasquill curves assume time-averaged motion in the vertical direction, which is not accounted for in our model.

#### st712e Dataset, 70 meters Downwind

We repeat the analysis for the st712e dataset for the same averaging times, as shown in Figure 4.10.

Once again, the plume is off-center in relation to the mean wind direction. Again, there is very little change between 600 and 1200 seconds. Also as before the CALPUFF GPI average compares somewhat reasonably to the Pasquill-Gifford curves. As a whole, the average is on the order of the classical result.

#### **st712c Dataset, 20 & 40 meters Downwind**

As we continue to move closer to the stack, agreement between the CALPUFF GPI and Pasquill-Gifford theory (somewhat unexpectedly) improves. We attribute this to a decrease in vertical dispersion as predicted by Pasquill-Gifford theory and a decrease in the traverse distance caused by an angular change in wind direction.

#### **st712e Dataset, 20 & 40 meters Downwind**

As with the st712e dataset, the agreement between Pasquill-Gifford theory and our model results improves.

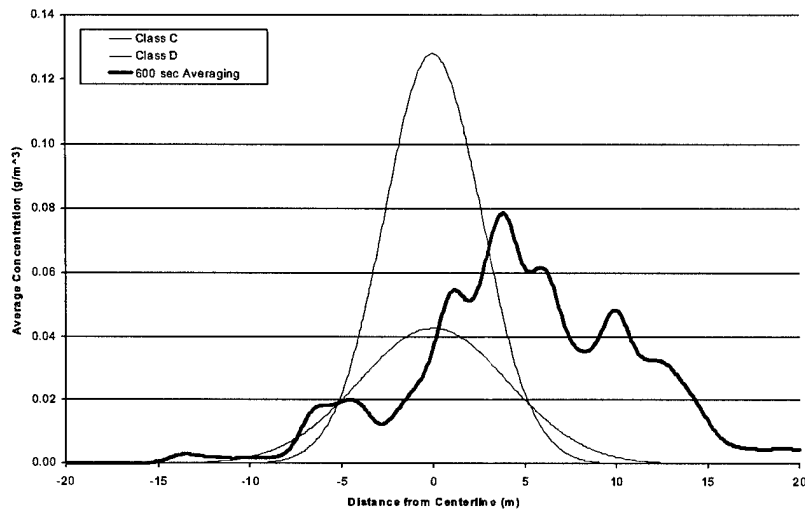


Figure 4.11: Average Concentration over 60 seconds for Test st712c at 40 meters Downwind

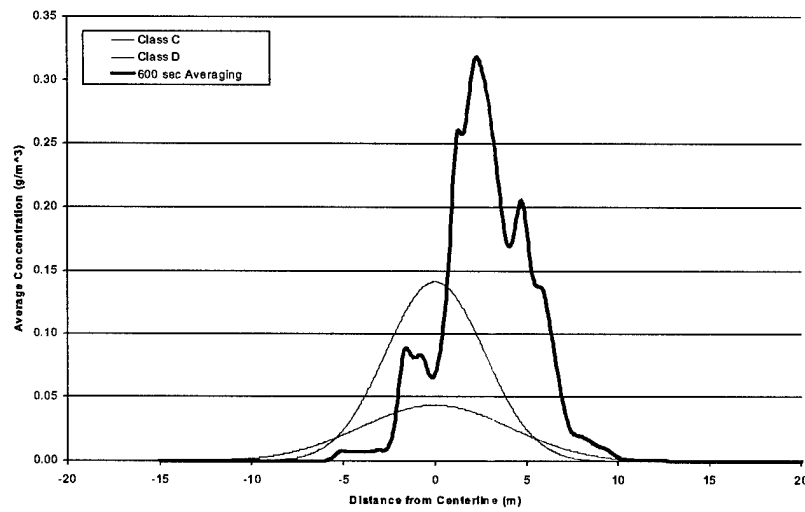


Figure 4.12: Average Concentration over 300 seconds for Test st712c at 20 meters Downwind

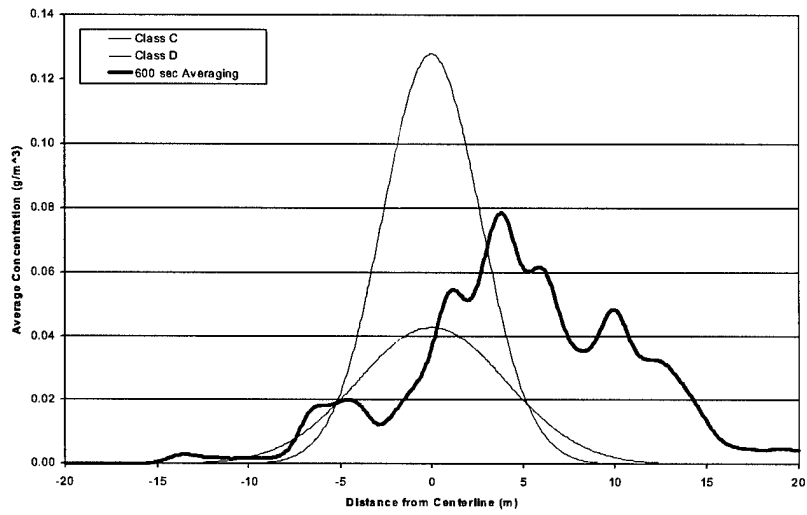


Figure 4.13: Average Concentration over 60 seconds for Test st712c at 40 meters Downwind

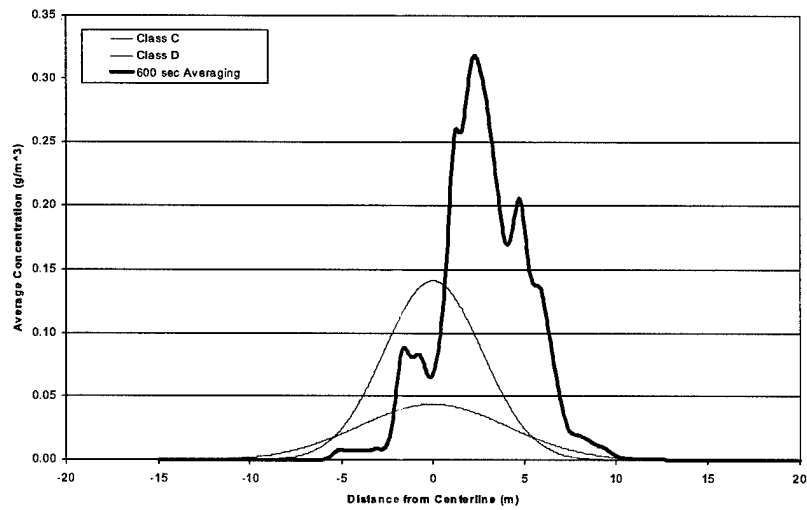


Figure 4.14: Average Concentration over 300 seconds for Test st712c at 20 meters Downwind

### 4.1.3 Validation Summary for CALPUFF Client

Despite the absence of field data for direct comparison, we have demonstrated to some degree the validity of the overall CALPUFF GPI model. The GPI client generates identical dispersion data to MIND, validating that the routines perform as intended and generate the proper values for the instantaneous theory of Dr. Peterson and her colleagues.

For moderate downwind distances (for SIG purposes), we appeal to the validation efforts in O'Neill (1996b) to demonstrate that the technique produces concentration time series that are similar in character to observed time series and that average concentrations are similar to the average values predicted by Pasquill-Gifford theory.

As SIG applications tend to require modeling at extremely short downwind distances, we have computed our own time-averaged results. These show that, as downwind distance decreases, our method diverges from the results predicted by traditional theory. However, we note that such short downwind distances are beyond the original scope of both the original Pasquill curves (Pasquill, 1961) and the curve fits commonly used (Scire et al., 1999).

## 4.2 Validation for Abramovich Client

We have not validated the results from the Abramovich client in any way. From the thermal truth images obtained from DIRSIG, in fact, we suggest that the Abramovich model is not very realistic. The hot area created by even a small vent is much larger than common sense would lead us to suspect.

The fundamental problem with this model is that it was developed for isothermal jets, i. e., jets of the same temperature as the ambient air. The flow in such jets is neutrally buoyant, and thus tends to behave in a laminar fashion; this is shown conceptually in Figure 4.15. In heated jets, on the other hand, the flow will tend to rise due to buoyancy, separating it from the roof. The thermal differences tend to induce turbulence more quickly than in

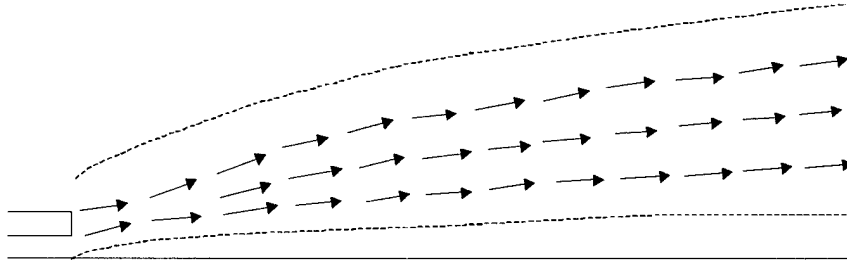


Figure 4.15: Conceptual Flow for Axial Isothermal Jet

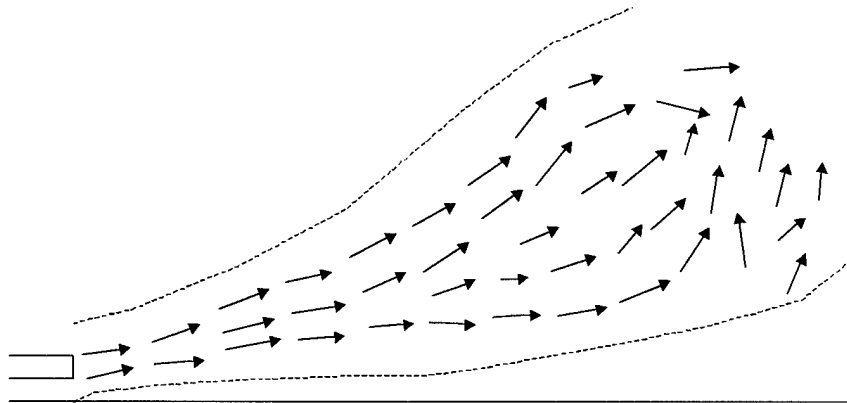


Figure 4.16: Conceptual Flow for Axial Heated Jet

isothermal flow through the induction of local velocity gradients. The result is likely a plume that separates from the surface and disperses more quickly than Abramovich theory suggests; this is conceptually shown in Figure 4.16.

Despite this severe caution, the client still serves a two-fold purpose. First, it is an example of a relatively simple GPI client that calculates temperatures from theoretical equations (as opposed to the CFD client, which performs only a blind interpolation); with development of a better theory, new equations could be added to the basic skeleton of the client.

Second, the general shape of the heated area is intuitively right. The initial region, in which the centerline velocity does not decay, causes the plume to widen out as the dispersion increases. In the subsequent main region, centerline decay causes the plume to narrow, producing an overall teardrop shape (which can be seen in Figure 4.23). By scaling the downwind distances in some manner and/or changing the dispersion relationships, the size of the teardrop could be scaled. This could be done either based on experiment or on a rough-order estimate. If these corrections can be applied, the speed advantage of the Abramovich client, with its simple, on-the-fly calculations, might make it more attractive than the CFD client.

### 4.3 Validation for CFD Client

As with the Abramovich client, we have performed no validation for the CFD client. The CFD model itself is well validated as it is intended to be a government-produced standard tool. Thus, insofar as the user correctly runs the code, checks the results, and takes care to use proper models (e. g., turbulence models) for the particular problem, the WIND results will be accurate.

In the case of our roof vent CFD solution, we are guilty of not strictly following our immediately preceding advice. The final solution is not, in fact, converged. The mass flow rate through the downstream end of the grid changes quickly at first (from the assumed initial condition, which is uniform flow throughout the grid) and then oscillates about a mean value, changing by about 15%. (The flow through the other grid boundaries also varies to conserve mass.) Tightening the limits by which the flow is allowed to change each

iteration did not damp out this oscillation. There are two likely explanations. First, the limits may need to be tightened further to force the solution to “creep up” on the steady-state value. Second, the proper solution may, in fact, not exhibit static stability, in which case the result is correct for a given instant in time.

From basic observation of the results, the solution is approximately correct. The vent flow enters vertically and is smoothly deflected by the freestream. The downstream velocity is higher than the freestream value due to the entrainment of the jet's air. This is shown in Figure 4.17. The edge of the data field nearest the observer is the centerline (symmetry) plane of the vent. The vent blows downward, and the freestream blows from right to left. The flow can be seen to curve as it is entrained by the freestream. Also visible is a shear between the vent air and the freestream, which is deflected downward only slightly.

Figure 4.18 shows the two-dimensional temperature distribution on the roof for our example simulation. The vent is a square centered at a downwind distance of three meters with a diameter of 0.5 meters. When rendering, the distribution is reflected about the centerline.

Figure 4.19 shows temperature contours in the vicinity of the vent. The edge of the contour volume closest to the observer is a plane of symmetry. The bottom of the volume is the roof. The vent, which is rectangular, is not drawn, but its position is noticeable at the top of the volume because of the constant temperatures in its vicinity. Both the highest and lowest temperatures are dark, while moderate temperatures are gray. (All of the hottest temperatures are near the vent. They are surrounded by the moderate temperatures, and the coolest temperatures are outermost.) The contour closest to the vent corresponds essentially with the vent temperature; as there are no contours within the area immediately under the vent, the air is at a nearly constant temperature. The asymmetric temperature distribution due to the freestream wind is visible.

We note that the maximum temperature for most downwind distances is not on the centerline but slightly outboard. This could be caused by an instability in the flow, but we

VELOCITY

0.006	MACH
0.00 DEG	ALPHA
$1.41 \times 10^{-5}$	Re
$1.30 \times 10^{-5}$	TIME
7x11x13	GRID 1
17x11x25	GRID 2
11x11x13	GRID 3
7x11x5	GRID 4
17x11x5	GRID 5
11x11x5	GRID 6

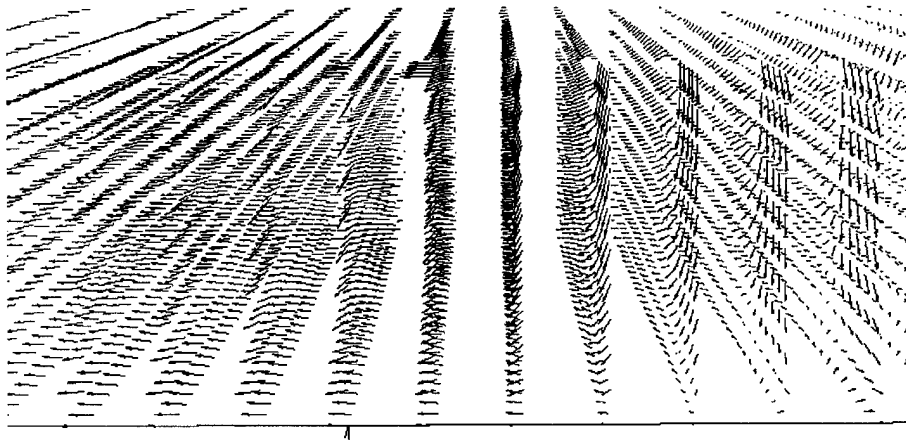


Figure 4.17: Velocity in Vicinity of Downward Vent

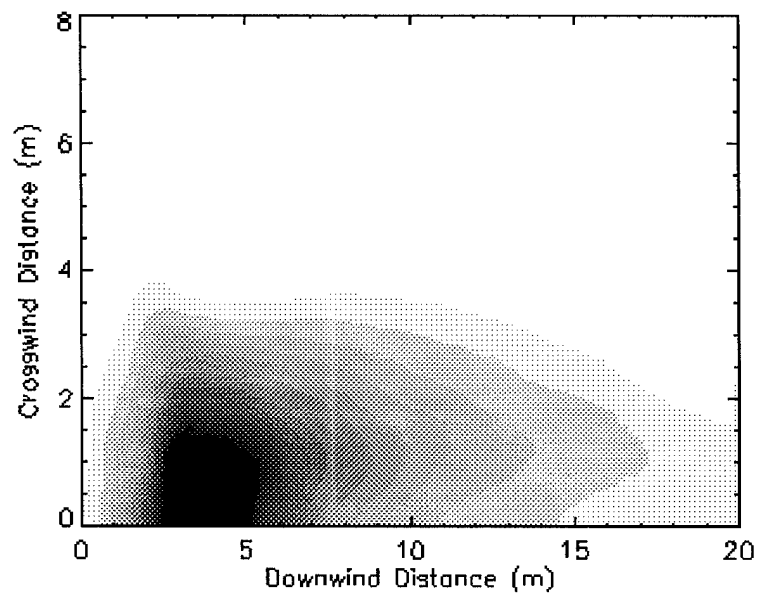


Figure 4.18: Roof Temperature Distribution for Downward Vent

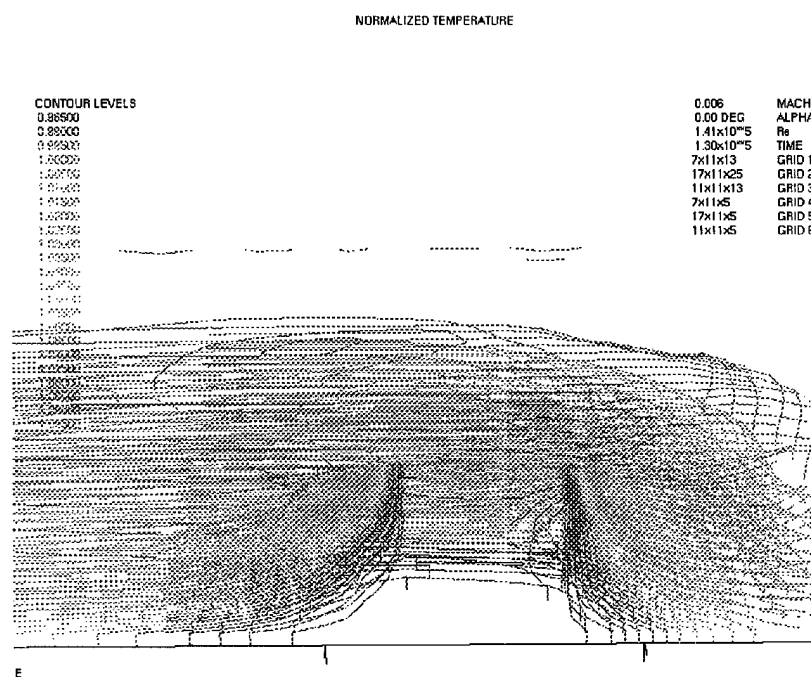


Figure 4.19: Air Temperature Distribution in Vicinity of Vent

suggest that it is more likely that this is caused by an improperly converged solution. This defect is not visible except at very high resolution, so its effect is not major in most cases. Developing an improved solution requires more complete monitoring of convergence information from WIND under a variety of solution parameters to determine which parameters most affect the character of parameter oscillations. This task would be best performed by someone specializing in computational fluid dynamics.

## 4.4 CALPUFF GPI Time Series Images

Our first result using the CALPUFF GPI client is a sequence of images in DIRSIG's standard "Nevada Test Site" scene. This scene is ideal for rendering a sequence of images because there are few objects in the scene so that render time is reasonably short. The scene was previously used by Kuo (1997) in his plume model development. We use the same scene, but our field of view and wind direction are different.

The wind history for this sequence is our simulated version of st712c, as shown in Figure 3.18. We have simulated images every one second for one minute and thereafter every ten seconds until five minutes total elapsed time. Images from zero, five, and ten seconds are shown in Figures 4.20 through 4.22.

These images clearly show the movement of the plume over time. When viewed as an animation, it can be seen that disturbances propagate outward. The plume appears a bit like a slack rope with a transverse wave moving along it.

## 4.5 Hawkeye Image

As an all-up test of our plume models, we have rendered a scene of the Eastman Kodak Hawkeye plant in Rochester, NY. The scene is centered on a factory stack that is emitting an  $\text{SO}_2$  plume. The wind sequence for this plume is the same as that used for the NTS scene (the "standard" simulation of dataset st712c), although the sequence is rotated overall. Also

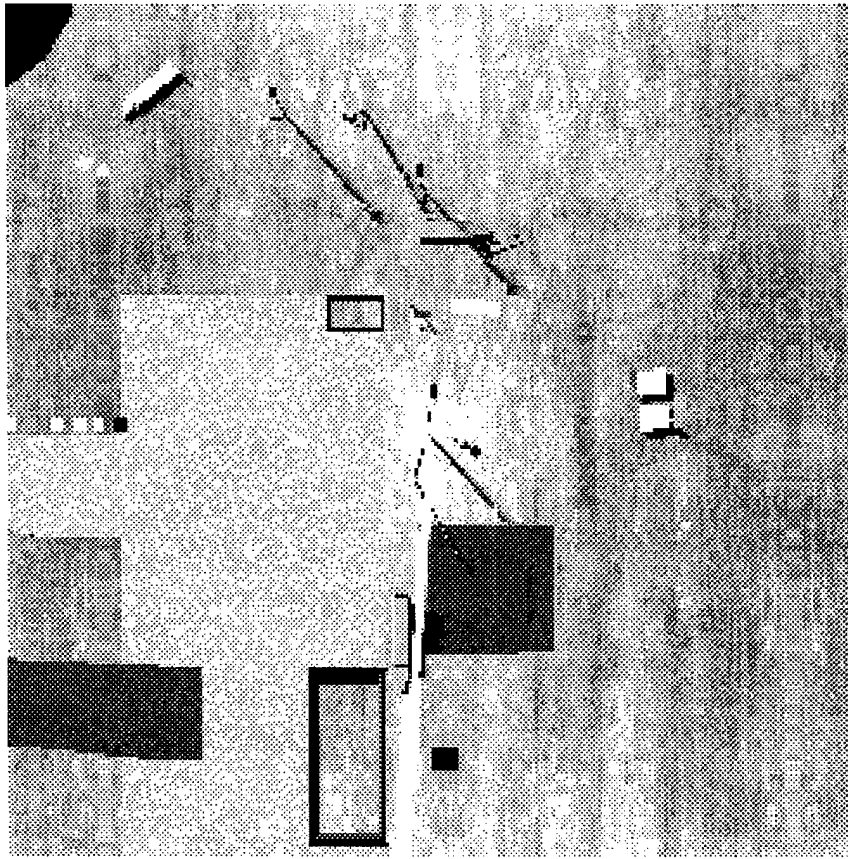


Figure 4.20: Nevada Test Site Plume at  $t = 0$  seconds

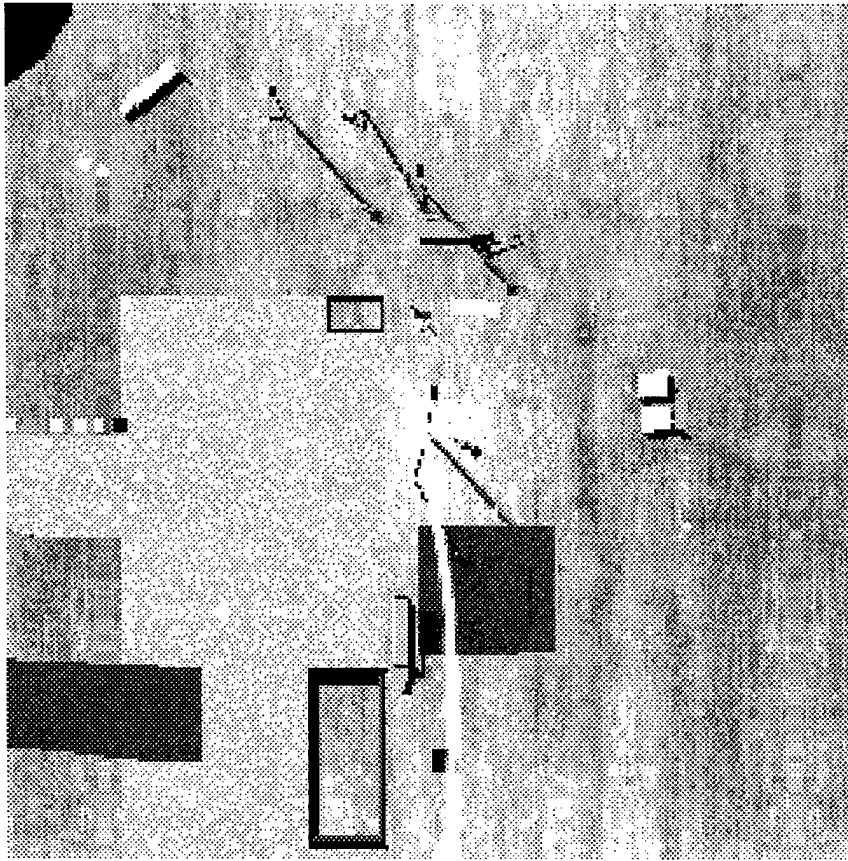


Figure 4.21: Nevada Test Site Plume at  $t = 5$  seconds

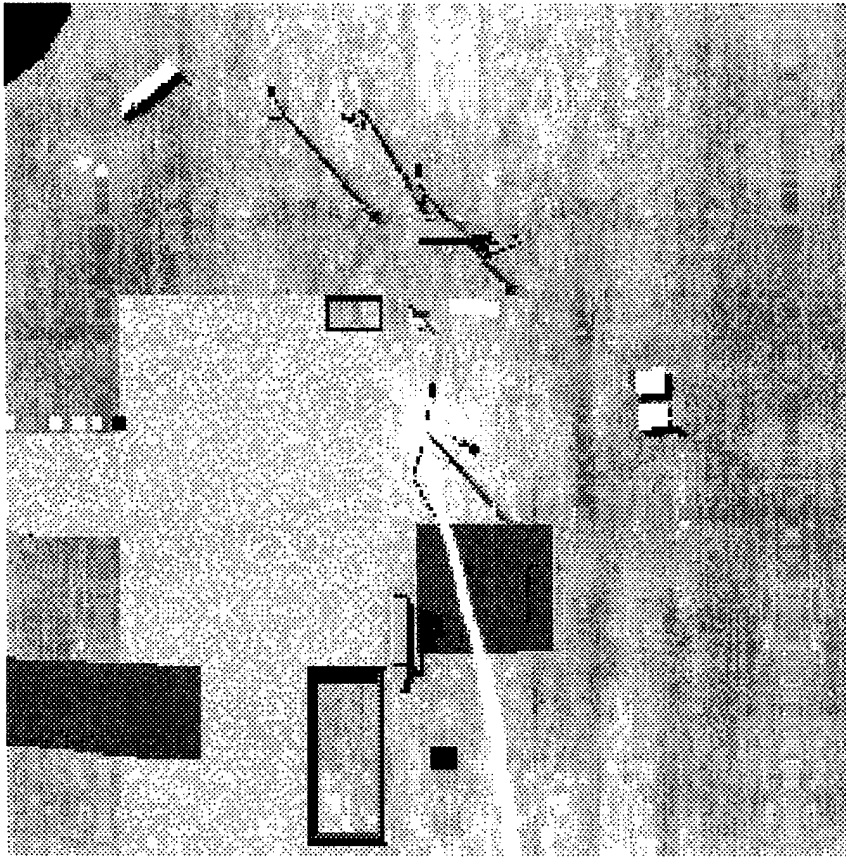


Figure 4.22: Nevada Test Site Plume at  $t = 10$  seconds



Figure 4.23: Hawkeye Image at Peak  $\text{SO}_2$  Emission

present in the image are two thermal vents, an Abramovich vent and a CFD-modeled vent. Both vents and the factory plume are visible in Figure 4.23. The Abramovich vent is clearly visible in the upper-left corner, and the CFD vent is seen as a small spot roughly on a line connecting the source of the Abramovich vent and the factory stack. The image conversion process has caused these images to appear more quantized in tone than the raw DIRSIG images are.

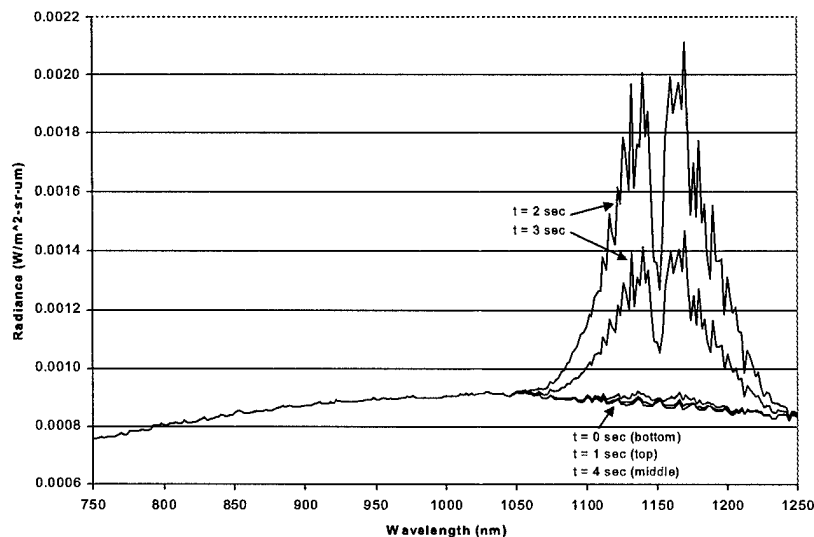


Figure 4.24: Radiance as a Function of Time for a Pixel Near Plume

#### 4.5.1 Spectral Signature Variability

The wavelength used for this image, 1186 nm, is the peak emission wavelength for  $\text{SO}_2$ . As the CALPUFF plume moves, the spectral radiance from a point near the plume changes. This effect is shown in Figure 4.24. When the plume is not over the point of interest (as at  $t = 0$  and  $t = 4$ ), the  $\text{SO}_2$  signature is not visible. When the plume is close to the point of interest ( $t = 2$  and  $t = 3$ ), the line structure of  $\text{SO}_2$  is highly visible. When the plume is more distant ( $t = 1$ ), the features are barely perceptible; if the sensor has poor signal to noise, this weak signal may not be distinguishable from the background.

Although it is not visible, the  $\text{SO}_2$  plume continues off the image to the left. The plume is also somewhat wider than it appears; although the concentration is measurable for some lateral distance, the temperature falls off fast enough that only the core emits appreciably. The dispersion at the stack exit has been set equal to 1.5 times the stack diameter; this produces the best match between the stack size and visible plume diameter.

The teardrop shape of the Abramovich plume is clearly visible. Also noticeable is the truncation of the Abramovich plume at the edge of the building. As each ray is traced, a simple height check is performed. If the height of the point of interest differs from the height of the vent base by more than one meter, no heating contribution is reported. It must be emphasized that this effect will not properly simulate impingement on a wall. If an Abramovich plume strikes a wall, raised portion of a roof, etc., the client will report that the portion of the wall within one meter of the vent elevation is at an elevated temperature, and the temperature elevation will abruptly cut off at more than one meter height difference. Furthermore, the temperature reported for the wall will be the temperature at the top of the boundary layer, so that there would be no vertical attenuation.

#### **4.5.2 Comparison to Experimental Data**

Although there are not a great many experimental images available, some experimental imagery of the Nevada Test Site (NTS) is available. We qualitatively compare it to the Hawkeye scene. In both, the plume is very bright near the stack. It quickly dims and is difficult to detect within about 10 stack diameters downwind. In both cases the plume spreads gradually without blooming. (We note that gas plumes do not necessarily behave similarly to the water vapor plumes we often see. The water in a plume from, e. g., a cooling tower is near saturation and thus condenses and evaporates with changing conditions, so that its visibility is not easily predictable.)

These images demonstrate that sensing of gas plumes must be done near the source; temperatures and concentrations, and thus emission, drop very quickly.

#### **4.5.3 Side-Looking views**

Although the overhead view is more applicable to satellite or airborne remote sensing of plumes, side-looking sensors are suited for ground-based monitoring of factories, experimental collections, etc. The side view also demonstrates the modeling of plume rise in our



Figure 4.25: Experimental Nevada Test Site Scene with Annotations

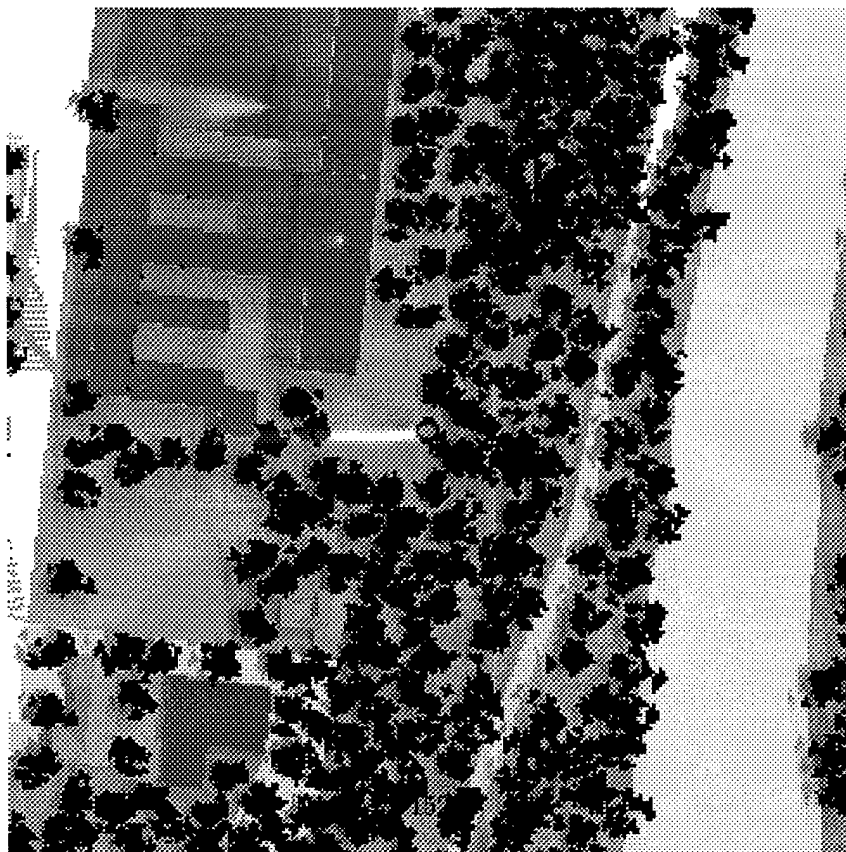


Figure 4.26: Hawkeye Scene



Figure 4.27: Side-Looking Hawkeye View from Kuo (1997)

model and highlights an improvement of our model over past work. An image from Kuo (1997) is shown in Figure 4.27.

Two similar view using our method are shown in Figures 4.28 and 4.29. Note that the size of our plume is a closer match to the stack size than that of Figure 4.27. The older method breaks the plume into finite regions; these regions are nested to give something of a gradual appearance to the plume, but the changes are noticeably discrete, and the plume ends at a finite distance downwind. In contrast, the radiance from our plume varies due to wind speed fluctuations, which cause the spacing between puffs to vary.

Figure 4.28 was simulated using a wind speed of 3.6 m/s, while Figure 4.29 uses a wind speed of 1.5 m/s. The effect of wind speed on the shape of the plume is readily apparent: in slower wind, the plume rises more vertically than it does with faster wind. Additionally, the variations in plume intensity are closer together when the wind is slower. (The standard deviation of the wind was not modified when the mean speed was changed; in reality, the standard deviation should be somewhat correlated with the mean speed, so



Figure 4.28: Side-Looking Hawkeye View Using CALPUFF GUI (3.6 m/s Wind)

that the fluctuations in slower wind would be less severe than shown here. As we have no experimental data regarding this relationship, we are unable to model it here.)



Figure 4.29: Side-Looking Hawkeye View Using CALPUFF GUI (1.5 m/s Wind)



## Chapter 5

# Conclusion

### 5.1 Summary

In this dissertation, we have developed two novel techniques for simulating plume-related phenomena. First, we have made possible the simulation of time-varying “directly sensed,” or factory, plumes. Although previous work has explored the direct sensing of plumes, our model features the following innovations:

- The dispersions used for these plumes reflect the actual instantaneous dispersions more closely than previous models have.
- The time-varying nature of the model enables not only the simulation of video imagery but also the simulation of other types of imagery and instruments that require relatively long imaging times (e. g., Fourier Transform Spectrometers, time-delayed integration, slowly moving line scanners).
- The motion of the plumes is governed by realistic random wind fields generated with statistics closely matched to field observations.
- The radiance emitted by plumes is calculated in a more accurate, step-wise process.

Second, we have simulated the heating of the environment by plumes. Our models incorporate the following:

- Warming effects are coupled with other thermal effects in a first-order manner.
- Three types of vents are available depending on user needs and available computer resources:
  - Transverse vents resemble small factory stacks and rise in a similar manner. Although often unimportant, the model highlights parallels between directly-sensed and indirectly-sensed plumes.
  - Axial vents discharge parallel to the freestream wind. Although not as accurate as the other models, the model runs very quickly.
  - Arbitrary geometries can be simulated by computational fluid dynamics (CFD). This option requires considerable effort and computer time, but makes possible the incorporation of a very wide range of phenomena.
- Wind speed variations from vents are also calculated for future use.

In addition to the preceding scientific and technical advances, we have also described the following practical and user-oriented developments:

- A key component of the new plume model is the EPA's state-of-the-art CALPUFF code. If enhancements are made to CALPUFF's puff-location techniques, they will be transparently added to our plume model.
- All plume models are implemented as clients to the new DIRSIG Generic Plume Interface (GPI). GPI clients execute their models independently of the main DIRSIG program. Communication is through a structured set of messages. Users can incorporate their own plume models into new clients, allowing them to be easily tested.
- Multiple plumes can now appear in the same image.

We have validated the plume models, when possible, against traditional plume modeling techniques, other related work, and experimental data. The models performed within the expected range. Long-term average results compare reasonably well to traditional methods.

Finally, we have computed and described sample images demonstrating the time-varying nature of plumes and the ability to simultaneously model multiple plume instances and/or phenomena.



## Chapter 6

# Suggested Further Development

Although we have demonstrated the merits and advancements of our new methods, there remain a few enhancements and modifications that would be beneficial prior to widespread release of the finished product.

### 6.1 Loose Ends

#### 6.1.1 CALPUFF Client

- **Remove hard-coded autoregression parameters**—Autoregression coefficients and the order of the AR model are hard-coded into the client. Recommend moving these to a file that is read during `gpi_init`.
- **Remove hard-coded wind upsampling**—The order of upsampling to be used when converting from the sequence generated by the autoregression to the wind sequence (which is the same as the the order of down-sampling used during model development to remove excessive autocorrelation from the original time sequence) and the magnitudes of the noise added during upsampling are hard-coded into the client. Recommend moving these to a file that is read during `gpi_init`.

## 6.2 New Functionality Requiring Minimal Research

### 6.2.1 CALPUFF Client

- **Automate execution of CALPUFF**—Due to time constraints, the code to automatically write the CALMET and CALPUFF input files has not been completed. The actual `printf` statements for the input files have been assembled and work, but the reading in and calculation of some variables remains, as does writing of the ancillary meteorological data files (surface wind, winds aloft, etc.). Some thought should be given to the translation to the CALPUFF coordinate system (which cannot handle negative coordinates). A more complex aspect of this topic is the addition to the CALMET and CALPUFF input files of dynamically generated elevation data from the DIRSIG database.
- **Add mixing lid to concentration calculation**—Currently, the plume reflects only from the ground and not from the mixing lid. The full functionality of Equation 2.27 could be added fairly easily. It may not be too important, though, as inversions lead to haze and poor remote sensing conditions.

## 6.3 New Functionality Requiring Significant Research and Testing

- **Convert CALPUFF puffs to subpuff-filled spheres**— Although we can modify the matching parameter to fit the plume to the stack, this is still an *ad hoc* adjustment. A gradual transition from a uniform distribution near the stack to a Gaussian distribution downstream can be accomplished through the subpuff method described in subsubsection 2.1.3. This may require significant experimentation to find the appropriate number of subpuffs and to ensure that there aren't any gaps near the stack exit.

- **Analysis of wind data**—Although the sampling rate is not quite high enough, the new rooftop weather station may provide data useful to better parameterizing wind characteristics. Knowledge of the variability in coefficients between and within stability categories is important.
- **Axial vent characterization**—This may be more a mechanical engineering topic than an imaging science topic. It would be valuable to determine through experimentation (either physical or CFD-based) how accurate the Abramovich model is when used on non-isothermal jets. Even if it is inaccurate, this problem geometry is probably well behaved enough in parameter space to use a lookup table approach to find solutions.

## 6.4 New Functionality Requiring Significant Research and Modification of Current Method

- **Puff tracking**—Because the current method incorporates spatial wind variation only as an aggregate effect through averaging, it is not well-suited to simulating discontinuities in plumes, wind shear, and other localized effects. An approach that individually emits and propagates puffs through time would be capable of simulating these discontinuities. The actual movement of puffs is simple, but the method requires a realistic wind field varying in space and time to produce plumes that are more realistic than those produced by the current method. Implementing this approach will require significant research into wind statistics and modeling. This area is discussed in more detail in Appendix G.



## Appendix A

# Rural Dispersion Coefficients

Taken from EPA, 1995b

$$\sigma_y = 465.11628(x) \tan(\theta)$$

$$\theta = 0.017453293[c - d \ln(x)]$$

Pasquill Stability Category	<i>c</i>	<i>d</i>
A	24.1670	2.5334
B	18.3330	1.8096
C	12.5000	1.0857
D	8.3330	0.72382
E	6.2500	0.54287
F	4.1667	0.36191

$$\sigma_z = ax^b$$

Pasquill Stability Category	$x$ (km)	$a$	$b$
A*	< 0.10	122.800	0.94470
	0.10 - 0.15	158.080	1.05420
	0.16 - 0.20	170.220	1.09320
	0.21 - 0.25	179.520	1.09320
	0.26 - 0.30	217.410	1.26440
	0.31 - 0.40	258.890	1.40940
	0.41 - 0.50	346.750	1.72830
	0.51 - 3.11	453.850	2.11660
	> 3.11		
B*	< 0.20	90.673	0.93198
	0.21 - 0.40	98.483	0.98332
	> 0.40	109.300	1.09710
C*	All	61.141	0.91465
D	< 0.30	34.459	0.86974
	0.31 - 1.00	32.093	0.81066
	1.01 - 3.00	32.093	0.64403
	3.01 - 10.00	33.504	0.60486
	10.01 - 30.00	36.650	0.56589
	> 30.00	44.053	0.51179
E	< 0.10	24.260	0.83660
	0.10 - 0.30	23.331	0.81956
	0.31 - 1.00	21.628	0.75660
	1.01 - 3.00	32.093	0.64403
	3.01 - 10.00	33.504	0.60486
	10.03 - 30.00	36.650	0.56589
F	> 30.0	44.053	0.51179
	< 0.20	15.209	0.81558
	0.21 - 0.70	14.457	0.78407
	0.71 - 1.00	13.953	0.68465
	1.01 - 2.00	13.953	0.68465
	2.01 - 3.00	14.823	0.54503
	3.01 - 7.00	16.187	0.46490
	7.01 - 15.00	17.836	0.41507
	15.01 - 30.00	22.651	0.32681
	30.01 - 60.00	27.074	0.27436
	> 60.00	34.219	0.21716

\* If the calculated value of  $\sigma_z$  exceeds 5000 m,  $\sigma_z$  is set to 5000 m.

\*\*  $\sigma_z$  is equal to 5000 m.

## Appendix B

# Urban Dispersion Coefficients

Taken from EPA, 1995b

Pasquill Stability Category	$\sigma_y$ (m)	$\sigma_z$ (m)
A	$0.32(x)(1.0 + 0.0004x)^{-1/2}$	$0.24(x)(1.0 + 0.001x)^{1/2}$
B	$0.32(x)(1.0 + 0.0004x)^{-1/2}$	$0.24(x)(1.0 + 0.001x)^{1/2}$
C	$0.22(x)(1.0 + 0.0004x)^{-1/2}$	$0.20(x)$
D	$0.16(x)(1.0 + 0.0004x)^{-1/2}$	$0.14(x)(1.0 + 0.0003x)^{-1/2}$
E	$0.11(x)(1.0 + 0.0004x)^{-1/2}$	$0.08(x)(1.0 + 0.0015x)^{-1/2}$
F	$0.11(x)(1.0 + 0.0004x)^{-1/2}$	$0.08(x)(1.0 + 0.0015x)^{-1/2}$

$x$  is in kilometers



## Appendix C

# Listing of CALPUFF GPI Parameters and Inputs

### C.1 Breakdown of Parameters

The parameters and inputs in the CALPUFF GPI code can be broken down into a few basic groups.

1. Stack geometry and effluent inputs
2. Release information
3. CALMET and CALPUFF inputs
4. AR Model parameters
5. MIND algorithm parameters
6. Ray tracing parameters

## C.2 Stack Geometry and Effluent Inputs

The parameters below are used to define the stack geometry and effluent types in the DIRSIG `.release` file. The typical values are rough order of magnitude dimensions.

Number	Units	Release file tag	calpuff_gpi variable	Typ val
1.1	STACK{LOCATION}	m	release->stack_location	As req (Note 1)
1.2	STACK{HEIGHT}	m	release->stack_height	15
1.3	STACK{DIAMETER}	m	release->stack_diameter	2
1.4	EFLUENTS{COMPOUND{NAME}}	N/A	N/A	As req
1.4	EFLUENTS{COMPOUND{WEIGHT}}	Nondim	release->gas_weights []	As req

Note 1: The stack location does *not* have to be identical to the stack location input into CALPUFF (parameter 3.9). The location in the release file may have negative coordinate values (i. e., the origin does not have to be at the corner of the scene), whereas the CALPUFF input requires positive values. The axes must be aligned, however. The necessary coordinate transformation is performed by the initiation portion of the GPI.

## C.3 Release Information

The release parameters are also contained in the DIRSIG `.release` file. They are all contained in the `RELEASE{}` tag at the end of the file. The parameters for each time are specified on a single line in the order shown in the table. All variables listed are actually arrays with elements for each of the time listed in the file. The reasonable values are roughly equivalent to the st712c data set.

Number	Parameter	Units	calpuff_gpi variable	Reasonable value
2.1	Time	TBD	release->time	Note 1
2.2	Exit velocity	m/s	release->exit_velocity	12.0
2.3	Exit temperature	K	release->exit_temperature	350
2.4	Air temperature	K	release->air_temperature	300
2.5	Wind velocity	m/s	release->wind_velocity	3.6 (Note 2)
2.6	Wind direction	deg	release->wind_direction	0
2.7	Atmospheric stability	N/A	Not currently used	
2.8	Release rates	g/s	release->gas_rates[]	20

Note 1: Interpolation of release values requires parameter entries before and after the time of interest. Therefore, it is recommended that the user always enter release parameters for a time of 0.0 and a time after the last time of interest. These parameters can be identical if the release properties are constant.

Note 2: For consistent results, the user must ensure that the wind velocity specified in the release file is the same as that used in the CALPUFF input file. Furthermore, the CALPUFF GPI client does not use time-varying wind speeds; only the average speed is calculated. Therefore, for clarity, the user should specify the same wind speed for all time inputs.

## C.4 CALMET and CALPUFF Inputs

The following inputs are in the CALMET and/or CALPUFF .cfg files. Other inputs are required to specify the number of meteorological stations used and the files containing meteorological data; such auxiliary inputs are not listed here. Furthermore, there are many relationships between parameters that must be checked for consistency. The “M” and “P” columns indicate whether a parameter is used in CALMET and CALPUFF, respectively.

Number	Parameter	CAL* tag	Units	M?	P?	Rec value
3.1	Number of grid points	NX & NY	Nondim	X	X	Note 1
3.2	Grid scale	DGRIDKM	m	X	X	0.01
3.3	Origin of coord sys	XORIGKM & YORIGKM	km	X	X	Note 2
3.4	Debug flag	LDEBUG	N/A		X	T
3.5	# of puffs to track	NPFDEB	Nondim		X	1 Note 3
3.6	Max travel dist	XSAMLEN	Grid units		X	0.01 Note 4
3.7	Max # of puffs to release	MXNEW	Nondim		X	20000
3.8	Max # of sampling steps	MXSAM	Nondim		X	1000 Note 5
3.9	Source parameters	SRCNAM	Various		X	Note 6

Note 1: NX or NY, the number of grid points, multiplied by DGRIDKM, the size of each grid square, must be larger than the region to be rendered in DIRSIG.

Note 2: The origin for the coordinate system is the location of the southwest corner of the grid in the Universal Transverse Mercator (UTM) coordinate system. The origin is not currently used, but the user must ensure that the location of the origin is consistent with the locations of any meteorological stations provided.

Note 3: Although we are interested in the locations of many puffs at an instant, we ask CALPUFF to write to the debug file the trajectory of only one puff through time.

Note 4: XSAMLEN is the distance that a puff will be allowed to move during a sampling step. A puff position is written to the CALPUFF debug file each sampling step. By setting it small (0.01 m in this case), we get a very high resolution plume track from CALPUFF.

Note 5: MXSAM is the maximum number of of sampling steps that will be used; it is intended to reduce computational time if XSAMLEN is very small. MXSAM should be set large so that this capping does not occur.

Note 6: Stack parameters are listed on a single line. The only ones important are XUTM and YUTM (first and second fields in the line), which are the location of the stack in kilometers referenced to the UTM grid. Care must be taken to ensure that the location lies within the computational grid. The CALPUFF location does *not* have to be identical to that in the release file (parameter 1.1). The relative spacing between the stack and any

terrain or other features must be preserved.

## C.5 AR Model Parameters

Many parameters in the CALPUFF GPI client deal with the autoregressive wind model. Some of the parameters are defined as preprocessor macros (`#define`), while others as variables. The names of the preprocessor macros are capitalized. The default parameters are for the st712c dataset.

Number	Parameter	Units	Description	Default value
4.1	<code>AR_DOWNSAMPLE</code>	Nondim	Factor by which wind data are downsampled prior to AR modeling	10
4.2	<code>AR_MODEL_ORDER</code>	Nondim	Order of AR model	4
4.3	<code>max_time</code>	s	Maximum elapsed time for which wind is generated	200
4.4	<code>target_sigmatheta</code>	deg	Desired standard deviation for wind direction	11.37
4.5	<code>sigmau</code>	m/s	Desired standard deviation for wind speed	0.812
4.6	<code>ubar</code>	m/s	Desired mean wind speed	3.6
4.7	Mean wind direction (no var)	deg	First elements of wind array initialized to mean	353.068
4.8	AR coefficients	Nondim	AR coefficients hard coded	As per Eq 3.8
4.9	<code>sigma_noise_direction</code>	deg	Noise added to wind direction during upsampling	0.5
4.10	<code>sigma_noise_speed</code>	deg	Noise added to wind speed during upsampling	0.5

## C.6 MIND Algorithm Parameters

After the wind sequence is generated the MIND algorithm is run. The following parameters are associated with this portion of the client.

Number	Parameter	Units	Description	Default value
5.1	<b>minr</b>	m	Minimum distance at which instantaneous diffusion is calculated (Note 1)	10
5.2	<b>numr</b>	Nondim	Number of distances at which instantaneous diffusion is calculated	500
5.3	<b>deltar</b>	m	Interval at which instantaneous diffusion is calculated (Note 2)	1.0

Note 1: **minr** must be at least two seconds downwind of the source as measured using the mean wind speed.

Note 2: The furthest distance at which the diffusion is calculated,  $\text{maxr} = \text{minr} + (\text{numr} - 1)\text{deltar}$ , must be farther from the source than the most distant CALPUFF debug puff.

## C.7 Ray Tracing Parameters

Only one user parameters is actually used in ray tracing. It is the matching criterion by which the plume dispersion is matched to the stack size. The stack diameter is *divided* by the constant, and the result is added to all the dispersions calculated by MIND.

Number	Parameter	Units	Description	Default value
6.1	Minimum dispersion (no var)	m	Added to dispersions calculated by MIND	1.0

## Calculation of $u_m$

The diagram illustrates a two-dimensional flow field. On the left, a vertical velocity profile  $u_n$  is shown. The flow is represented by streamlines, with two specific streamlines labeled  $M$  and  $N$ . A curved obstacle is positioned in the flow, with a velocity profile  $u_m$  indicated at its base. The horizontal distance from the origin  $O$  to the obstacle is  $x_n$ , and the horizontal distance from the obstacle to the right boundary is  $x_H$ . The vertical distance from the  $x$ -axis to the top of the obstacle is  $b$ . The origin  $O$  is marked on the  $x$ -axis.

Pertinent variables are:

Quantity	Definition	Explanation
$b$		Jet thickness from top of boundary layer to top of jet
$b_0$		Initial jet height (stack height)
$b_{\Pi}$	$\frac{b_{\Pi}}{b_0}$	Jet thickness at $x_{\Pi}$
$\bar{b}_{\Pi}$	$\frac{b_{\Pi}}{b_0}$	Nondimensionalized jet thickness at $x_{\Pi}$
$m$	$\frac{u_H}{u_m}$	Local velocity ratio
$m_0$	$\frac{u_H}{u_0}$	Initial velocity ratio (freestream to jet)
$u$		Air velocity at an arbitrary point
$u_m$		Air velocity just outside of boundary layer
$u_H$		Freestream wind velocity
$u_0$		Vent gas velocity
$\bar{u}$	$\frac{u}{u_H}$	Nondimensionalized velocity
$x$		Downwind distance
$x_{\Pi}$		Beginning of main region; point at which $u_m$ begins to decrease
$\bar{x}_{\Pi}$	$\frac{x_{\Pi}}{b_0}$	Nondimensionalized beginning of main region
$\delta$		Boundary layer thickness
$\delta_{\Pi}$		Boundary layer thickness at $x_{\Pi}$
$\bar{\delta}_{\Pi}$	$\frac{\delta_{\Pi}}{b_0}$	Nondimensionalized boundary layer thickness at $x_{\Pi}$
$\chi$	$1/m$	Inverse of velocity ratio (for convenience)
$\chi_0$	$1/m_0$	Inverse of initial velocity ratio (jet to freestream)

First, the parameters of the problem are established:  $b_0$ ,  $u_H$ ,  $u_0$ . From the velocities,  $m_0$ , and  $\chi_0$  are found trivially.

Next, the properties at the beginning of the main section ( $x_\Pi$ ) are found.

$$\bar{b}_\Pi = \frac{b_\Pi}{b_0} = \frac{1 - m_0}{0.316 (1 - m_0) (1 + 0.425m_0) + 0.0875} \quad (11.43)$$

$$\bar{\delta}_\Pi = \frac{\delta_\Pi}{b_0} = \frac{0.1}{1 - m_0} \bar{b}_\Pi \quad (11.42)$$

$$\bar{x}_\Pi = \frac{x_\Pi}{b_0} = \frac{\bar{b}_\Pi + \bar{\delta}_\Pi - 1}{0.22 \frac{1-m_0}{1+m_0} (0.584 - 0.134m_0)} \quad (11.45)$$

From these,  $b_\Pi$ ,  $\delta_\Pi$ , and  $x_\Pi$  can be found.

Next, we evaluate a function of  $\chi_0$  that arises from an integration of  $db/b$ .

$$F_1(\chi_0) = \frac{\chi_0 - 1}{(\chi_0 + 0.4085)(\chi_0 - 0.8475)^{1.8}} \quad (\text{Between 11.33 and 11.34})$$

Although we will not use it in our analysis,  $b$  for a given location may be found, if desired, from  $\frac{b}{b_\Pi} = \frac{F_1(\chi)}{F_1(\chi_0)}$ . Note, when consulting Abramovich, that the exact equation given just above is not used subsequently. Instead, an approximation  $F_2(\chi) \approx F_1(\chi)$  that involves only integer powers is used; the approximation is Equation 11.34.

After determining the location and characteristics of the beginning of the main section, the properties at any downwind location can be determined. Our goal is to find  $u_m$  vs.  $x$ . First, a value for  $m$  (which must be greater than  $m_0$ ) and thus for  $u_m$ . Then,  $x$  can be found from

$$x = x_\Pi + \frac{b_\Pi}{0.22 \frac{F_1(\chi_0)}{0.85}} [F_3(\chi) - F_3(\chi_0)] \quad (11.38 \text{ (trivially modified)})$$

where

$$F_3(\chi) = 61 \ln \left( \frac{\chi - 1}{\chi - 0.8475} \right) + 0.925 \ln \left( \frac{\chi - 1}{\chi + 0.4085} \right) + \frac{10.02}{\chi - 0.8475} + \frac{1.47}{(\chi - 0.8475)^2} + \frac{0.374}{\chi + 0.4085} \quad (11.39)$$

By iterating over a series of values for  $m$  and solving for  $x$  for each, we derive the relationship between  $x$  and  $u_m$ . An example for  $m_0 = 0.2$  and  $b_0 = 1.0$  is shown in Figure D. This

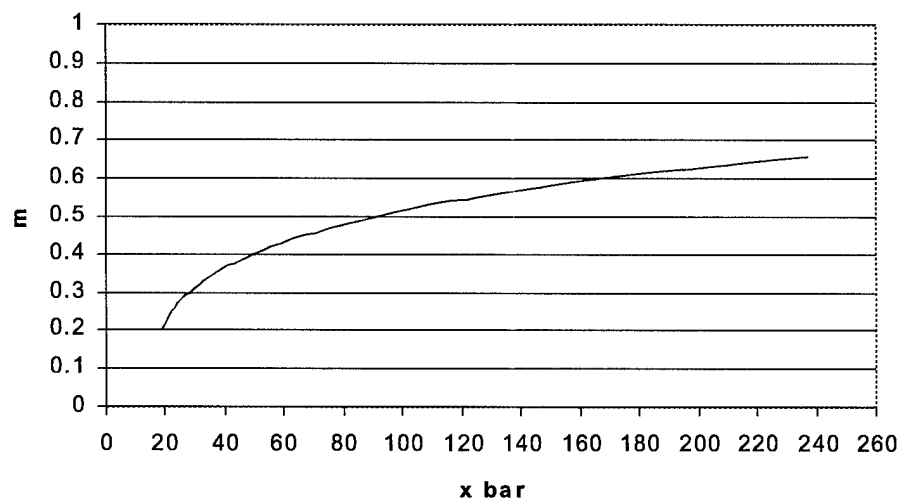


Figure D.1:  $m$  vs.  $x$  for  $m_0 = 0.2$

is the same case as the bottom solid curve in Figure 11.6 of Abramovich. When the  $F_2$  approximation is used, the curves are identical; they are slightly different when the exact  $F_1$  is used as in Figure D.

## Appendix E

# Procedure for Calculating AR Coefficients in R

Although Minitab is probably easier for overall analysis of time series and graphing of the results, analysis in R (R Development Core Team, 2001) is quicker and easier to automate for wind sequences that are well-modeled by autoregressive models (e.g., `st712c` and `st712e`).

### E.1 R Packages

As a general-purpose data processing language, R has no intrinsic time series analysis routines. The routines we have used are contained in two packages. The first, `ts`, is distributed with R and is maintained by the core R team. It includes routines used to create time series objects for analysis by other functions and several AR analysis routines. The second, `tseries`, must be downloaded separately. It includes alternate AR routines.

Library packages must be loaded before use. This is done using the `library` command:

```
library(ts)
library(tseries)
```

## E.2 Reading Time Series

Data must be read into a time series object before it can be analyzed in R. The `ts` library includes a function, `read.ts`, for this purpose. The series to be read should be put in a plain text file, with the data occupying a single column. A sample use of the function is:

```
dirts<-read.ts("10sec712cdir.puv",header=FALSE)
```

The “`header=FALSE`” argument suppresses interpretation of the first row as variable names. (We have read them from separate one-column files, but speed and direction could be read from a single file with headings.)

## E.3 Calculation of ACF and PACF

As mentioned in Subsection 2.3.2, the behavior of the Autocorrelation Function and Partial Autocorrelation Function indicate the type of ARIMA model appropriate for a time series. The `acf` and `pacf` functions plot the ACF and PACF along with significance bounds:

```
acf(dirts)
pacf(dirts)
```

## E.4 AR Modeling

Several procedures are available for conducting the actual autoregressive analysis. Various algorithms are available within these procedures to calculate the coefficients. We have found the `ar` and `arma` routines most useful.

### E.4.1 ar Routine

The `ar` routine is the more flexible of the two. It can use the Yule-Walker (the default), ordinary least squares, and maximum likelihood methods. With the ordinary least squares method the data can be “demeaned” (the mean subtracted from each point prior to analysis) and an intercept term can be fitted. The `ar` procedure also uses the Akaike Information Criterion to automatically select the order of the model.

```
ar.ols(dirts)
ar.ols(dirts,intercept=TRUE)
```

### E.4.2 arma Routine

Although the `ar` routine is more flexible, it does not report error bounds or significance levels for the model coefficients. The `arma` routine can do this. The routine uses the maximum likelihood method. The order of the model must be specified as a two-dimensional vector containing the order of the AR model and the order of the MA model.

```
arma(dirts,order=c(4,0))
summary(arma(dirts,order=c(4,0)))
```

The “summary” causes the significance of the coefficients to be reported.

### E.4.3 Example Session

An example session analyzing the data from the `st712c` dataset is shown here.

```
> library(ts)
> library(tseries)

'tseries' version: 0.7-0

'tseries' is a package for time series analysis with emphasize
on non-linear modelling.
See 'library(help=tseries)' for details.
```

```
> dirts<-read.ts("10sec712cdir.puv",header=FALSE)
> pacf(dirts)
```

The PACF is shown in Figure E.1.

```
> ar.ols(dirts,demean=FALSE,intercept=TRUE)
```

```
Call: ar.ols(x = dirts, demean = FALSE, intercept = TRUE)
```

```
Coefficients:
```

```
      1      2      3      4
0.4554 0.1046 0.1056 0.1496
```

```
Intercept: 65.18 (17.49)
```

```
Order selected 4  sigma^2 estimated as 65.82
```

```
> summary(arma(dirts,order=c(4,0)))
```

```
Call: arma(x = dirts, order = c(4, 0))
```

```
Model: ARMA(4,0)
```

```
Residuals:
```

```
      Min      1Q   Median      3Q      Max
-26.0807 -5.5484  0.2553  4.9188 25.1874
```

```
Coefficient(s):
```

	Estimate	Std. Error	t value	Pr(> t )	
ar1	0.45417	0.05716	7.945	2e-15	***
ar2	0.10440	0.06264	1.667	0.095585	.
ar3	0.10109	0.06260	1.615	0.106355	
ar4	0.14992	0.05717	2.622	0.008730	**
intercept	67.15283	17.37581	3.865	0.000111	***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Fit: sigma^2 estimated as 66.05,  Conditional Sum-of-Squares =
19483.37,  AIC = 2118.46
```

Although not exactly the same, the significant coefficients from these two procedures are very close to the results from Minitab in subsubsection 3.2.3.

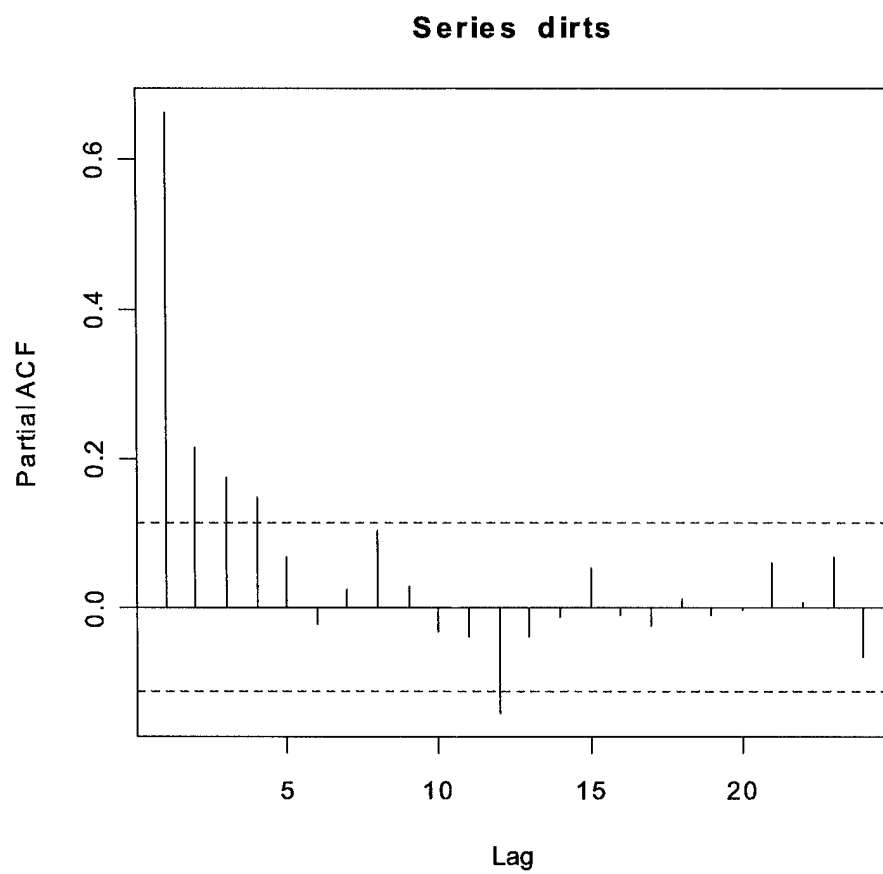


Figure E.1: Plot of PACF for Dataset st712c from R



## Appendix F

# Procedure for Using WIND CFD Results

This appendix outlines the procedure we have developed and implemented for incorporating results from the WIND CFD code into DIRSIG imagery. The procedure will be different if another CFD code is used, although the basic problem flow will be similar, and a very similar GPI client can be used if results can be exported as a simple list of the values of variables at each grid point.

### F.1 Required Software

#### F.1.1 WIND CFD Code

The WIND CFD code is a product of the NPARC Alliance, an organization formed by the teaming of NASA's Glenn Research Center and the U.S. Air Force's Arnold Engineering and Development Center. The organization is "dedicated to the establishment of a national, applications oriented computational fluid dynamics (CFD) capability, centered on the NPARC computer program." The original NPARC Alliance computer code is one of the three major components of the current WIND code. The homepage for the Alliance

is located at <http://www.arnold.af.mil/nparc>.

The WIND code is available to all U.S.-owned companies, universities, and government agencies; for non-governmental agencies, an export license is required. More information can be found at the NPARC home page.

The software is available for a wide variety of computer and processor combinations. Binary executable versions are available for most combinations, and the source code is also available. We have experience running the pre-compiled binaries under Linux on Intel processors, and have successfully built binaries for the Sun (we were unable to run the pre-compiled binaries due to an apparently hard-coded run-time library dependence). Compiling and installing the software is beyond the scope of this document; we will hereafter assume that the user has successfully run WIND and has worked through the *case4* tutorial described in the User's Manual (NPARC Alliance, 2001), which is, as of August, 2001, available at <http://www.grc.nasa.gov/www/winddocs/user/index.html>.

### **F.1.2 PLOT3D Visualization Code**

PLOT3D is a CFD visualization and plotting package developed by the NASA Langley Research Center. It is not strictly required for our method, but highly recommended, as it is the most straightforward way to display WIND's input and output to ensure correctness and reasonableness. PLOT3D can be obtained from <http://www.nas.nasa.gov/cgi-bin/software/start>. Execution of a license agreement is required.

## F.2 Method for Using WIND Results in DIRSIG

The steps involved in creating and using WIND CFD data are:

Step 1: Create grid file

Step 2: Create data file

Step 3: Execute WIND code

Step 4: Export results

Step 5: Modify cfd\_gpi

First, WIND's input files must be created. Two files are required. The grid (.cgd) file specifies the grid geometry and boundary condition types, and the data (.dat) file defines the specifics of the flow conditions and various options.

### F.2.1 Create grid file

WIND requires grid data in the Boeing Common Grid Format. Various tools are provided to assist the user in converting data to and from this format.

Although an ASCII version of a Common File is available, the files are quite long and not particularly user-readable. We have chosen, therefore, to use a modified version of the procedure used in the WIND User's Guide (NPARC Alliance, 2001) in the *case4* example. This method writes the data into the .xyz format used by PLOT3D, which is then converted to a .cgd file using the *cfcnvt* WIND utility.

We were unable to successfully read formatted binary files into *cfcnvt*, and thus use the unformatted binary format, similar to the *case4* example. We first write a C program that defines the grid geometry and outputs (x,y,z) coordinate locations as ASCII text. This text can easily be read by humans for error checking and can be exported to other platforms. The basic C program for our baseline roof vent geometry is *3dventbig.mesh.c*.

As the .xyz file is read by FORTRAN READ statements in PLOT3D, we write the actual .xyz file using identically-formatted FORTRAN WRITE statements to ensure that

the file is written correctly. The programs `trans.f` (for two-dimensional geometries) and `trans3d.f` (for three-dimensional geometries) take input from the C programs and write the `.xyz` file using the `WRITE` statements.

Using our method, the user modifies an appropriate C file to revise the geometry and compiles it. The `.xyz` file is then generated by piping the output of the C program through the FORTRAN-based translator; an example command line would be `3dventbig.exe | trans3d.exe`. The output from the translator, `trans.xyz`, can then be renamed.

Once the `.xyz` file has been written, it can be translated to a `.cgd` file using option 11 in `cfcnvt`, similar to the procedure used in the *case4* example.

The `.cgd` file as output by the translator contains only the geometry information; the user must subsequently define boundary condition information using the WIND utility `gman`. The procedure is again similar to that for the *case4* example. We have found it convenient to generate a command script file for `gman` which can then be used to quickly redefine all boundary conditions. Example scripts for many of our cases are available in their directories, titled `bc.jou`. The major differences between the various scripts are in the locations of the arbitrary inflow regions; most of the other boundary conditions do not change.

(We have found, for the Sun platform, what may be a bug in `gman`. When coupling the zones together on a fresh `.cgd` file (using the command `automatic couple face zone all`), `gman` crashes with a segmentation fault. The command and all subsequent commands seem to execute normally when `gman` is run again for the same file.)

### F.2.2 Create Data File

The data file contains flow specifics for a given case; for example, it contains the freestream flow properties, which are applied to all boundaries specified in `gman` as being of freestream type, and the properties for arbitrary inflow regions. It also contains options for turbulence models and various solver options. Details of the various commands are found

in the WIND User's Guide (NPARC Alliance, 2001). In many cases, our sample `.dat` files may be usable as is, or perhaps with modifications to the arbitrary inflow properties.

Some discussion of the appropriate boundary condition for the roof plane is necessary. For the most accurate results, we could account for the boundary layer buildup on the roof. This would be done by specifying the boundary as a "viscous wall" in WIND or by manually curving the bottom plane of the grid according to the boundary layer thickness and specifying the boundary as an "inviscid wall." Our sample analysis does neither; the simple heat transfer relations already present in DIRSIG do not account for boundary layer growth. For simplicity and CFD runtime savings, we chose to model the roof plane as an inviscid wall. Although unrealistic, this choice matches reality as DIRSIG knows it, as there is no boundary layer buildup on the wall.

The `.cgd` and `dat` files are the only required files. An `.mpc` file is also used if the solution is to be run in parallel on multiple machines. We have successfully run WIND in parallel using the `ssh` (Secure Shell) method for communication between machines.

### F.2.3 Execute WIND Code

After the input is defined, the WIND code is run in a manner similar to that for the *case4* example. If the CFD grid is large, the solution may take a considerable length of time; depending on the load from other users, we have found that the *3dventbig* problem (the geometry used for our roof vents) running in parallel on six machines takes between twelve and twenty-four hours to complete, depending on the load from other users.

### F.2.4 Export Results

Exporting WIND results is simple using the WIND utility `cfpost`. In a manner similar to that in the *case4* example, we specify that output is to be for all zones, but only for the  $j = 1$  plane, which forms the "roof" (the bottom of the computational domain). We then output the velocity and temperature results using the "list" command in `cfpost`. Our

example scripts export the data in the normal format, which is more easily human-readable, and the raw format, which is a sequential listing only of temperatures. The raw format is used directly by the DIRSIG GPI, but it is recommended that the normal format be kept with the raw format as a guide to grid point locations if necessary.

The results from the `.cfl` results file can also be converted into a PLOT3D `.q` file in a manner similar to the *case4* example so that the data can be viewed graphically in PLOT3D for evaluation.

### F.2.5 Modify `cf_d_gpi`

The final step in the process is to modify the `gpi` client, `cf_d_gpi`, to read the data file. This is necessary because, at present, the grid geometry is hard-coded. The sizes (dimensions and locations) of the zones are calculated by the `gpi` client. The geometry can be easily calculated if the program lines used to generate the mesh in the appropriate portions of the `.mesh.c` file are copied to `cf_d_gpi.c`. The `.raw.T` and `.raw.V` (temperature and velocity) files must also be moved to the directory from which DIRSIG will be run. Once the code has been copied, DIRSIG can be run invoking the `cf_d_gpi` client.

## Appendix G

# Puff Tracking

Although our method is an improvement over previous methods in that it can simulate some dynamic plume effects, it is not capable of modeling all phenomena. In particular, it will not allow the plume to become spatially discontinuous. Furthermore, the method is not easily adaptable to incorporating such effects. Fully modeling these phenomena, we suggest, requires a rigorous tracking of all puffs through a spatio-temporally varying wind field. Such an approach is largely incompatible with CALPUFF, incompatible with the plume centerline location of the MIND algorithm (MIND instantaneous diffusions are still usable), and requires a much more complex time series analysis of the wind. Although implementing any of these techniques is beyond the scope of our work, we describe in general an approach to achieving a puff tracking modeling technique in order to guide future research.

### G.1 Limitations of MIND Method

As discussed in subsubsection 2.2.5, the MIND algorithm uses averaging to approximate the aggregate effect of wind fluctuations on puffs as they travel downstream. This method is computationally efficient and is adequate in cases where the plume moves smoothly, i. e., when each puff approximately follows the previous one. When this is the case, the fact that

our method actually rotates the CALPUFF debug puffs from side to side is not apparent. This assumption of smoothness is not valid where wind shear or turbulence cause the plume to become discontinuous transversely to the mean wind direction, and our method does not propagate such discontinuities downstream. As mass rates are propagated downstream (the mass of each debug puff is set at each time step according to the release rate in effect at the appropriate time of release), discontinuities along the wind direction can be simulated on an *ad hoc* basis by varying the gas release rate. A method similar to the propagation of mass downstream could be used to propagate a fixed lateral or vertical break downstream or to introduce one in a manual fashion into a single image, but this improvised method would not automatically simulate the variable character that such breaks should exhibit.

## G.2 Overview of Puff Tracking Approach

The concept of a puff tracking model is simple to understand: puffs are emitted at a certain rate (sufficient to prevent gaps in a constant wind case) from a source. Based on a simulated wind velocity at the source, each puff is moved in the wind direction for a short length of time. At each time step a new puff is emitted and the previous puffs are again moved in the direction of the wind at their most recent location. This process is conceptually shown in Figure G.1. In the figure, puffs are emitted every  $\Delta t$  seconds, and the wind speed experienced at time  $i\Delta t$  by puff  $j$  is denoted by  $\vec{u}_{i,j}$ . Once the puff field is established, raytracing proceeds in exactly the manner used for our method.

## G.3 Wind Modeling Requirements for Puff Tracking

The example shown in Figure G.1 assumes that the wind varies with both space and time. This is essential for puff tracking.

If the wind field varies only temporally (e. g., if an experimental wind sequence is used across the entire problem domain), the relative spacing between puffs does not change with

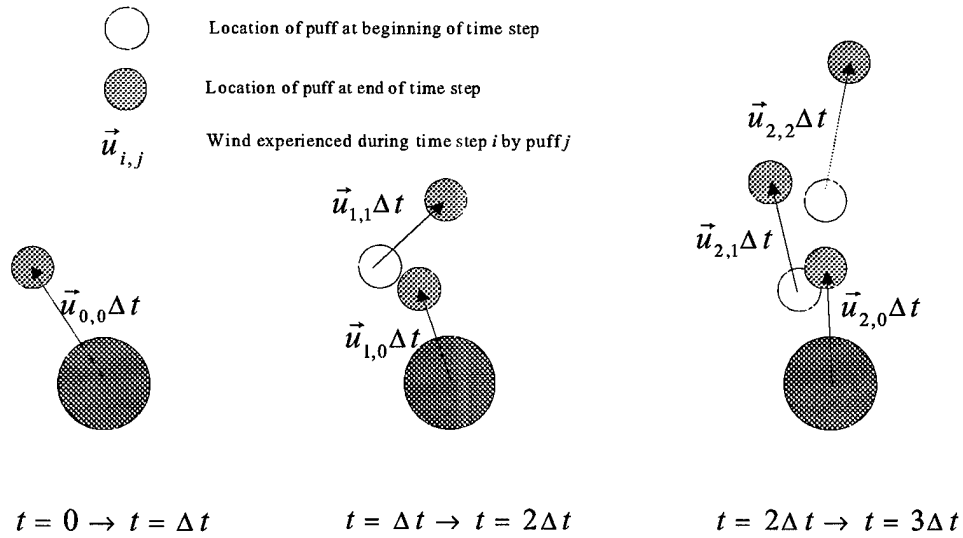


Figure G.1: Illustration of Puff Tracking

time; this effect, shown in Figure G.2, produces a continuous plume with jagged structure. This jaggedness will be reduced somewhat with distance as the size of the puffs grows, but the spacing between puffs will not change. This method does have the small advantage that it will capture variations in wind speed at any frequency (which is not true of the MIND model since major variations occur only with every second of downwind travel), but this advantage seems to be small compared to the problem of static puff spacing.

Alternatively, a wind field that varies only spatially would be incapable of simulating a time series; the plume would appear the same at all times. Thus, both spatial and temporal variability are required to model plumes adequately. The MIND method provides this by using variable smoothing times based on distance on a single wind sequence, thus introducing spatial variability but in a way that induces smoothness. (It would be possible to use a MIND-derived wind sequence with puff tracking, but the resulting plume would be smooth and generally similar in character to those from our normal method.)

To see a clear benefit from a puff tracking approach, a realistically-varying wind field

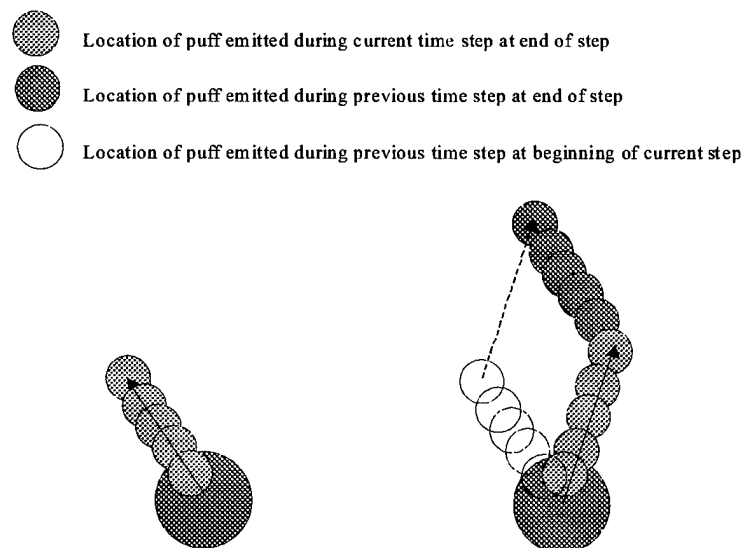


Figure G.2: Illustration of Puff Tracking with Spatially-Constant Wind

is needed. Unlike with MIND-smoothed wind, the variability in this wind field should not be a function of distance from the source; statistics such as the standard deviation, while varying locally, should probably not have any systematic variations across the field.

Once this wind field is available, puff tracking becomes a (large) matter of bookkeeping. As each puff is emitted, the wind it experiences in a time step is retrieved and the puff is moved a small distance accordingly. This process is repeated for “past” time steps until enough puffs have been emitted to form the plume at  $t = 0$  and then continued as needed to simulate the plume at future times. This tracking is likely to be quite time consuming.

## G.4 Eulerian and Lagrangian Wind Formulations

There are two fundamental ways to examine and model the wind field. The Eulerian approach measures the wind at fixed locations, as with receptor-based measurements. The Lagrangian formulation, on the other hand, measures properties from the perspective of a

particle moving in the flow; it is experimentally very difficult to measure as the point of measurement continually varies. In general, the autocorrelation and other wind statistics will differ depending on which perspective is used, with theoretical relationships between the two (Hanna, 1979; Blackadar, 1997).

If the wind is modeled only in the Eulerian approach, three-dimensional data must be simulated (two dimensions of space plus time) for both the speed and direction. This data must either be pre-generated or generatable on the fly. However, the overall correlation characteristics in each of the spatial and temporal dimensions will likely be the same through the entire wind cube.

Modeling the wind in a Lagrangian fashion reduces the dimensionality of the data to two for speed and two for direction: one dimension specifies the time of release of the puff (or some other puff index reference) and the other dimension specifies time. Thus, fewer data are required. The spatial dimensions are implicit in the formulation; the puff location can be calculated as the vector sum of all previous motions, and using Lagrangian statistics ensures that the next motion is appropriate for wherever the particle ends up. These motions are correlated with recent motions of the puff and with motions of puffs emitted at nearly the same time.

It seems likely that this reduction of dimensionality comes with another price: the autocorrelation functions in the two dimensions may not be independent of each other. In particular, we suggest that the motions of puffs emitted around the same time will become less correlated with travel time. While close to the stack the puffs will tend to move through the same turbulence cells. As time (and distance) progress, small variations between paths that puffs emitted at nearly the same time take will tend to move some puffs into different cells, and they will thus begin to experience different motions, reducing the correlation between subsequent predicted motions. Correctly modeling this effect may be difficult as the necessary models are nonlinear (though it may be possible to generate the data in piecewise-linear blocks with discrete changes in ARIMA parameters between blocks).

## G.5 Summary and Recommendations

If a satisfactory method for modeling multi-dimensional wind can be found, using it to generate plumes via puff tracking should lead to more realistic plumes. In the same way that our one dimensional AR process is capable of producing sudden random shifts in the wind direction, the multi-dimensional model should produce a wind field containing both temporal and spatial (shear) effects. As mentioned earlier, this process is likely to be quite resource intensive.

The first step in implementing a puff tracking approach is a search for any other work using ARIMA models to generate multi-dimensional wind; without a proper wind field, puff-tracked plumes will either be smooth, as our current plumes are, or suffer from immediately apparent artifacts. If the foundation for wind field generation does not exist, creating one is probably a doctoral-level task in statistics, meteorology, or mechanical engineering.

As with our one-dimensional data, the model parameters must be determined through analysis of experimental data. Methods for estimating Lagrangian statistics based on Eulerian measurements are available, allowing wind to be measured at fixed locations but applied to moving puffs.

The experimental data should be multi-dimensional to fully determine the correlations in each dimension. Although it might be possible to assume that nearby positions experience wind with identical statistics to a measured position, this is equivalent to assuming that turbulence is homogeneous in a region, which would fail to capture shear characteristics. Thus, wind data from an array of receptors is likely necessary. This data would be difficult to capture, requiring an investment in equipment and a methodical and likely lengthy collection campaign to obtain enough data to enable reasonable confidence bounds on the parameters.

In summary, then, properly implementing a tracking approach requires in-depth knowledge of wind processes in order to produce a result that is clearly superior to our method.

Although this would be a significant achievement, the level of effort required may make it unattractive. Better value may be obtained in the end from tweaking our current model to manually add discontinuities and to propagate these discontinuities in the same way that puff mass is propagated. With some sort of "mutation" of the discontinuities with time, the result may be adequate for many purposes.



# Bibliography

G. N. Abramovich. *The Theory of Turbulent Jets*. The M.I.T. Press, Cambridge, Mass., 1963.

*9th Joint Conference on Air Pollution Meteorology*, 1996. American Meteorological Society.

Milton R. Beychok. *Fundamentals of Stack Gas Dispersion*. Self-published, 3rd edition, 1994.

Alfred K. Blackadar. *Turbulence and Diffusion in the Atmosphere*. Springer, Berlin, 1997.

Bruce L. Bowerman and Richard T. O'Connell. *Time Series Forecasting: Unified Concepts and Computer Implementation*. Duxbury Press, Boston, second edition, 1987.

George E.P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, revised edition, 1976.

Jens Breckling. *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*. Number 61 in Lecture Notes in Statistics. Springer-Verlag, Berlin, 1989.

Gary A. Briggs. Plume rise. Technical Report TID-25075 (NTIS), U.S. Atomic Energy Commission, 1969.

Gary A. Briggs. *A Simple Model for Bent-Over Plume Rise*. PhD thesis, The Pennsylvania State University, 1970.

- Gary A. Briggs. Some recent analyses of plume rise observation. In *Proceedings of the 2nd International Clean Air Congress*, pages 1029–1032, New York, 1971. Academic Press.
- Gary A. Briggs. Diffusion estimation for small emissions. In *Environmental Research Laboratories, Air Resources Atmosphere Turbulence and Diffusion Laboratory 1973 Annual Report*, number USAEC Report ATDL-106. U.S. National Oceanic and Atmospheric Administration, 1974.
- Gary A. Briggs. Plume rise predictions. In Haugen (1975), pages 59–111.
- Gary A. Briggs. Plume rise and buoyancy effects. In *Atmospheric Science and Power Production*, number DOE/TIC-27601, pages 327–366. U.S. Department of Energy, 1989.
- Environmental Protection Agency. Industrial Source Complex (ISC) dispersion model user's guide. Technical Report EPA-450/4-79-030, U.S. Environmental Protection Agency, Research Triangle Park, NC, 1979.
- Environmental Protection Agency. User's guide for the Industrial Source Complex (ISC3) dispersion models, volume i—user instructions. Technical Report EPA-454/B-95-003a, U.S. Environmental Protection Agency, Research Triangle Park, NC, 1995a.
- Environmental Protection Agency. User's guide for the Industrial Source Complex (ISC3) dispersion models, volume ii—description of model algorithms. Technical Report EPA-454/B-95-003b, U.S. Environmental Protection Agency, Research Triangle Park, NC, 1995b.
- Environmental Protection Agency. A comparison of CALPUFF with ISC3. Technical Report EPA-454/R-98-020, U.S. Environmental Protection Agency, Research Triangle Park, NC, 1998.
- Environmental Protection Agency. Meteorological monitoring guidance for regulatory modeling applications. Technical Report EPA-454/R-99-005, U.S. Environmental Protection Agency, Research Triangle Park, NC, 2000.

- F. A. Gifford. Statistical properties of a fluctuating plume dispersion model. *Advances in Geophysics*, 6:117-137, 1959.
- F. A. Gifford. Atmospheric dispersion models for environmental pollution applications. In Haugen (1975), pages 35-58.
- F. A. Gifford. Turbulent diffusion-typing schemes: A review. *Nuclear Safety*, 17(1):68-86, 1976.
- James Halitsky. Transverse jet plumes: Final report. Technical Report NYO-3673-5, New York University, Department of Meteorology and Oceanography, Geophysical Sciences Laboratory, 1970. Prepared for the U.S. Atomic Energy Commission Under Contract No. AT(30-1) - 3673.
- James Halitsky, June 2000. Private correspondence.
- James A. Halitsky. A method for estimating concentrations in transverse jet plumes. *Air & Water Pollution International Journal*, 10:821-843, 1966.
- James A. Halitsky. Temperatures and concentrations in heated plumes. *Atmospheric Environment*, 2:419-422, 1968.
- James A. Halitsky. A jet plume model for short stacks. *JAPCA Notebook*, 39(6):856-858, 1989.
- Steven R. Hanna. Some statistics of lagrangian and eulerian wind fluctuations. *Journal of Applied Meteorology*, 18:518-525, 1979.
- Steven R. Hanna, Gary A. Briggs, and Rayford P. Hosker. Handbook on atmospheric diffusion. Technical Report DOE/TIC-11223, U.S. Department of Energy, 1982.
- Duane A. Haugen, editor. *Lectures on Air Pollution and Environmental Impact Analyses*, Boston, 1975. American Meteorological Society.

- S. Didi Kuo. *Synthetic Image Generation of Factory Stack and Water Cooling Tower Plumes*. PhD thesis, The Rochester Institute of Technology, Rochester, New York, August 1997.
- Charles Mazzone and Holly Peterson. An investigation of probability distribution for concentration fluctuation data. In *9th Joint Conference on Air Pollution Meteorology* Ame (1996), pages 122–123.
- J. L. McElroy. A comparative study of urban and rural dispersion. *Journal of Applied Meteorology*, 8(1):19ff., 1969.
- J. L. McElroy and F. Pooler. The St. Louis dispersion study. Technical Report Report AP-58, U. S. Public Health Service, National Air Pollution Control Administration, 1968.
- Minitab. Minitab 13, 2000. URL <http://www.minitab.com>.
- The NPARC Alliance. WIND user's guide. Technical report, NASA Glenn Research Center and Air Force Arnold Engineering Development Center, 2001. URL <http://www.grc.nasa.gov/www/winddocs/user/index.html>.
- Susan O'Neill. Development and testing of a model for instantaneous plume dispersion. In *9th Joint Conference on Air Pollution Meteorology* Ame (1996), pages 364–367.
- Susan M. O'Neill. The Meandering INSTantaneous Diffusion (MIND) model. Master's thesis, Montana Tech of The University of Montana, Butte, Montana, December 1996b.
- Frank Pasquill. The estimation of the dispersion of windborne material. *The Meteorological Magazine*, 90:33–49, 1961.
- Frank Pasquill. The dispersion of material in the atmospheric boundary layer—the basis for generalization. In Haugen (1975), pages 1–34.
- Holly Peterson, Patrick Ballard, and Brian Lamb. A new lagrangian approach to studying instantaneous plume dispersion and concentraion fluctuations. In *11th Symposium on*

- Boundary Layers and Turbulence*, number 6.17, pages 140–143. American Meteorological Society, 1995.
- Holly Peterson and Brian Lamb. An investigation of instantaneous diffusion and concentration fluctuations. *Journal of Applied Meteorology*, 34:2724–2746, 1995.
- Holly Peterson and Brian Lamb. A tracer approach to investigating instantaneous plume diffusion and turbulence. In *9th Joint Conference on Air Pollution Meteorology* American Meteorological Society (1996), pages 37–40.
- Holly Peterson, Brian Lamb, and David Stock. Interpretation of measured tracer concentration fluctuations using a sinusoidal meandering plume model. *Journal of Applied Meteorology*, 29:1284–1299, 1990.
- Holly Peterson, Dione Mazzolini, Susan O'Neill, and Brian Lamb. Instantaneous spread of plumes in the surface layer. *Journal of Applied Meteorology*, 38:343–352, 1999.
- Holly Peterson, Susan O'Neill, and Brian Lamb. A simple approach for estimating short-term peak concentrations with time-averaged models. In *12th Symposium on Boundary Layers and Turbulence*, number 5.4, pages 112–113. American Meteorological Society, 1996.
- R Development Core Team. *An Introduction to R*, 2001. URL <http://www.r-project.org>.
- Hermann Schlichting. *Boundary-Layer Theory*. McGraw-Hill Book Company, New York, seventh edition, 1979.
- Joseph S. Scire, David G. Strimaitis, and Robert J. Yamartino. A user's guide for the CALPUFF dispersion model (version 5.0). Technical report, Earth Tech, Inc., Concord, Mass., 1999.
- D. Bruce Turner. Workbook of atmospheric dispersion estimates. Technical Report AP-26, U.S. Environmental Protection Agency, 1970.

David C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, La Cañada, Calif., 1994.

D. J. Wilson and G. Winkel. The effect of varying exhaust stack height on contaminant concentration at roof level. *ASHRAE Transactions*, 88(2):515–533, 1982.

David J. Wilson. Effect of vent stack height and exit velocity on exhaust gas dilution. *ASHRAE Transactions*, 83(1):157–166, 1977.

David J. Wilson. Flow patterns over flat-roofed buildings and application to exhaust stack design. *ASHRAE Transactions*, 85(2):284–295, 1979.

Hubert Wood, November 2000. Private conversation.